TESI DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

DOTTORATO DI RICERCA IN
INGEGNERIA ELETTRONICA E DELLE TELECOMUNICAZIONI

IMAGE FORGERY DETECTION AND LOCALIZATION

DAVIDE COZZOLINO

Il Coordinatore del Corso di Dottorato
Ch.mo Prof. Daniele RICCIO

Il Tutore
Ch.mo Prof. Giovanni POGGI

A. A. 2014–2015

“We need at least two points of view
to have a prospect to ...”

Contents

List of Figures **vii**

Introduction **1**

1 PRNU-based localization of small-size forgeries **7**
1.1 Introduction 7
1.2 Background 9
1.3 Methodology 12
1.4 Experimental evaluations 16
1.5 Conclusions 18

2 Efficient dense-field copy-move forgery detection **21**
2.1 Related work 21
2.2 Proposed method 24
2.2.1 Computing a dense NNF 24
2.2.2 Post-processing based on dense linear fitting 30
2.2.3 Feature Extraction 34
2.3 Experimental evaluations 38
2.3.1 Fine tuning of the proposed method 39
2.3.2 Comparison with the state of the art 45
2.4 Conclusions 50

3 Feature-based approach for forgery localization **53**
3.1 Introduction 53
3.2 Proposed method 55
3.2.1 Feature extraction 55
3.2.2 Detection/identification 56
3.2.3 Localization 57
3.3 Preliminary tests on tampering detection 58

3.3.1 Camera-based detection 58
3.3.2 Processing-based detection 59
3.4 Tampering localization experiments 60
3.4.1 Camera-based localization 60
3.4.2 Processing-based localization 61
3.4.3 Performance comparison 61
3.5 Conclusions 64

Conclusion **67**

A The First IFS-TC Image Forensics Challenge **69**
A.1 Tool based on machine-learning 71
A.1.1 Detection based on machine-learning 73
A.1.2 Localization based on machine-learning 74
A.2 Tool based on block-matching 76
A.3 Tool based on camera sensor noise 78
A.4 Decision fusion 82
A.5 Results and Conclusions 83

List of Figures

1 Examples of celebrities before-after the photo editing [1]. . . . 2

2 Right: a digital image forgery of an Iranian missile test appeared on the front page of many major newspapers in 2008. A missile was digitally added to the image in order to conceal a missile on the ground that did not fire. Left: the pristine image used to create the forgery. [2]. 2

3 In September 2012, the National Review published a cover photo in which the original slogan "Forward" was replaced with the word "Abortion" [2]. 3

4 Three forgeries drawn from the Forensics Challenge dataset [3]. From left to right: forged image, mask of counterfeiting and original image. 4

1.1 Scheme of PRNU-based forgery localization techniques. . . . 10

1.2 Correlation field and its prediction for pristine image. 12

1.3 Scheme of guided filtering [55]. 15

1.4 An application of guided filtering to denoise a no-flash image under the guidance of its flash version. [55] 16

1.5 ROCs obtained with boxcar and guided filtering with forgeries of size: 48×48, 64×64, 96×96, and 128×128 pixels. 17

1.6 Sample results. From left to right, original and forged image, correlation field predicted, and computed by boxcar and guided filtering. 19

2.1 Examples of copy-move forgeries. The genuine images on the left and the forged images on the right. 22

2.2 Offset propagation. Initially (left), most offsets are wrong (red); a random good offset (green) propagates along the scanning order (center) until the whole copy-moved region is covered (right). 26

2.3 Offsets vary linearly over the copy-moved region in the presence of rotation (left), rescaling (center), or both (right). 29

2.4 Predictor geometry for direct scanning. Blue pixels are used to build various zero- and first-order predictors for the offset of pixel s (red). 30

2.5 Post-processing steps: (a) original forged image, (b) magnitude of offsets, (c) median filtering, (d) fitting error $\epsilon^2(s)$ (dB), (e) thresholding of $\epsilon^2(s)$, (f) final mask. 33

2.6 Radial profiles of some CHTs: Zernike (left), PCT (center), FMT (right). 34

2.7 Examples of rectangular (a), polar (b) and log-polar (c) sampling grids. 37

2.8 Three forged images with different levels of activity from the FAU database. From top: smooth, rough, structured. 38

2.9 Three forged images from the GRIP database with different levels of activity. From top: smooth, mixed, textured. 40

2.10 Pixel-level F-measure curves for the proposed PM-based technique (Zernike-polar feature) for different values of N_{it}. . . . 44

2.11 Pixel-level F-measure curves for the proposed PM-based technique ($N_{it} = 8$) with different features. 44

2.12 Pixel-level F-measure curves for the proposed technique, the baseline reference, and intermediate versions. 46

2.13 Image-level F-measure curves for the proposed (PM-ZM-polar) and reference techniques. 48

2.14 Pixel-level F-measure curves for the proposed (PM-ZM-polar) and reference techniques. 48

2.15 Image-level F-measure curves for some selected techniques on textured images. 49

2.16 Pixel-level F-measure curves for some selected techniques on textured images. 49

2.17 Forgery detection masks for some images of the GRIP database. From top to bottom: forged images, masks output by Christlein2012, masks output by the proposed method. Green indicates correct detection, false alarms are in red. From left to right: noise with normalized std 0.02, JPEG compression with Q=60, rotation with $\theta = 45^o$, rescaling with $\alpha = 1.145$, occlusive rigid copy-move. 51

3.1 Graphic scheme of aggreation procedure. 57
3.2 Camera-based (left) and processing-based (right) detection performance using a Canon EOS 450D as target camera. . . . 58
3.3 Camera-based localization performance (Canon EOS 450D on the left and Nikon D200 on the right). 60
3.4 Performance comparison (Canon EOS 450D on the left and Nikon D200 on the right). 62
3.5 Processing-based localization performance (Canon EOS 450D on the first row and Nikon D200 on the second row). From left to right: blurring, JPEG compression, resizing, rotation. 63
3.6 Forgery localization results for some selected examples. From left to right: forged image, PRNU correlation index field and color-coded detection mask, proposed aggregation map and color-coded detection mask. Green indicates correct detection, false alarms are in red. 65
3.7 Forgery localization results for some selected examples. From left to right: forged image, PRNU correlation index field and color-coded detection mask, proposed aggregation map and color-coded detection mask. Green indicates correct detection, false alarms are in red. 66

A.1 A training image with its ground truth and an example residual image. 72
A.2 Scores (top) and AUC (bottom) for all models. 74
A.3 Two training fake images, their SDH map and the color coded detection mask. Green indicates correct detection, false alarms are in red. 75
A.4 Four training images with copy-move forgeries, their ground truth, and detection maps output by our method. 77
A.5 Steps adopted in PRNU-based tool. 78
A.6 Number of images belonging to the clustered sets. 80

A.7 Two training fake images, correlation maps and color-coded detection masks. Green indicates correct detection, false alarms are in red. 81
A.8 A training fake image, its correlation map, its PatchMatch-based map, and the final color-coded mask. 82
A.9 Flow chart of the combination strategy. 83
A.10 Four images from the test set and their output masks. 86

Introduction

Digital Photography is having a rapid and ever growing diffusion in recent years, since it allows anyone to take an arbitrary number of good quality images, quickly and at no cost, and to store them easily on a large number of digital supports, or share them on the Internet. At the same time, with the wide availability of advanced image editing tools (e.g. Adobe Photoshop, Gimp), modifying a digital photo, with little or no obvious signs of tampering, has become also very easy and widespread. For example, photo editing is largely used in entertainment in order to improve image appearance, like in the examples of figure 1. However, besides these benign cases, there are also malicious ones. Photo editing can be used in journalism to modify the meaning of an image, influencing the opinion of the readers, as shown in figures 2 and 3, or even in a court of law [2], to falsify evidences and possibly modify the final rulings. Therefore, the problem arises of establishing the integrity of digital images used as precious pieces of information in several fields of life.
Over the past dozen years, this problem has been faced by the scientific community in the field of *digital image forensics* and different methods have been proposed to validate the content of a digital image. However, the many existing approaches, which comprise in turn a very large and ever growing number of individual detection techniques, testify both the interest towards this problem and its complexity. In particular, *image forgery detection* deals with the techniques used to prove whether a digital image is pristine or has been tampered with, while *image forgery localization* concerns the techniques used to localize the forged region in a tampered digital image. Recently, a large number of these approaches have been proposed in the literature under a variety of scenarios.

A first category comprises *active* techniques for image authentication, which are based on the use of watermarks [54] and signatures [112]. In the first case, the watermark is embedded into the image (possibly originating some small distortions), while in the latter, the signature is attached to the

Figure 1: Examples of celebrities before-after the photo editing [1].

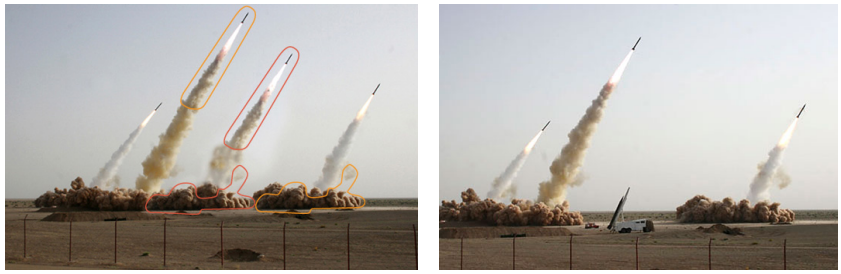


Figure 2: Right: a digital image forgery of an Iranian missile test appeared on the front page of many major newspapers in 2008. A missile was digitally added to the image in order to conceal a missile on the ground that did not fire. Left: the pristine image used to create the forgery. [2].

image as a side information. Although these methods are very effective, they can be applied only when the digital source is protected at the origin, which is probably a minority of the cases of interest.

Therefore, there has been a steadily growing interest on *passive* techniques which retrieve traces of manipulations from the image itself, with no need of collaboration on the part of the user. These techniques in fact are based on the observation that each step of the digital image life cycle (from the acquisition process in the camera to its recording in a compressed format and its subsequent editing) leaves a trace in the image, that can be extracted by the algorithm in order to reveal the tampering [88]. As noted by [39], these techniques use one of the following four approaches.

Figure 3: In September 2012, the National Review published a cover photo in which the original slogan "Forward" was replaced with the word "Abortion" [2].

Pixel-based techniques analyze the correlation between pixels either directly in the spatial domain or in some transformed domain. For example some methods are based on blur inconsistency [96], or on revealing traces caused by resampling [63], which is a necessary operation whenever the forgery needs to be rotated or rescaled by a certain factor. Other techniques are specifically tailored to detect duplicated regions in the image (as in figure 2) [29, 62, 6].

Format-based methods, instead, exploit the usual adoption of some lossy compression scheme, like JPEG. Several forgery detection techniques rely on the traces left by multiple compression. In fact, when a JPEG image is modified and saved again in JPEG format, specific artifacts appear as a result of the multiple quantization processes, suggesting the presence of some forms of tampering [74, 21, 14].

Camera-based techniques take advantage of peculiar traces left during the acquisition phase, related to lens characteristics [106, 46], the color filter array (CFA) pattern [90, 40], or the sensor array [77, 20]. These features, specific of any different camera models, or even of individual cameras, can be used as image signatures and exploited for forgery detection.

Phisics/geometric-based methods study higher-level inconsistencies, such as the lighting of objects, shadows, or geometric features (dimension, position, etc.) of objects present in the scene [60, 61, 76].

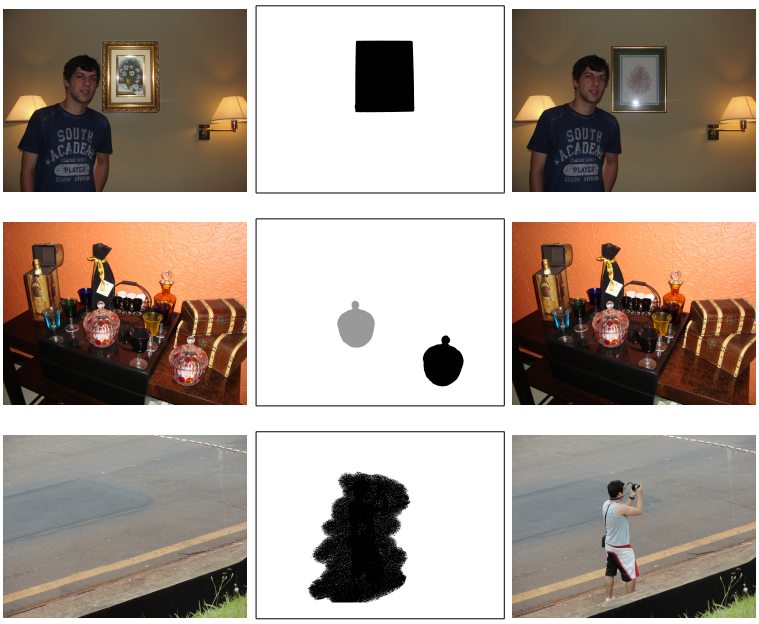


Figure 4: Three forgeries drawn from the Forensics Challenge dataset [3]. From left to right: forged image, mask of counterfeiting and original image.

Note that, each method works in very specific hypotheses which limit its applicability to a limited class of forgeries, and no ultimate all-encompassing solution exists to the image forgery detection and localization problems. Moreover, techniques proposed in the scientific literature are not always correctly validated. Sometimes they are tested on very specific proprietary datasets in favorable conditions, casting doubts on their actual performance in more general scenarios. In addition, often neither the source code nor an executable version are made available to guarantee reproducible research. Therefore, it is difficult to assess objectively such methods and figure out their performance in real-world applications [34].

Driven by these considerations, in 2013 the IEEE Information Forensics and Security Technical Committee (IFS-TC) launched the First Image Forensics Challenge [3], focused on image detection and localization (more detail in the Appendix). This was a means to foster new research on these topics and to compare the results of competing techniques on a large, publicly available and well designed dataset. In figure 4 there are three examples drawn from the

Forensics Challenge dataset. The first example is a *splicing* where an object has been added in the image. The second one is a *copy-move forgery* where a region of the image has been duplicated. The last example is based on inpainting operations to remove an object from the image. One of the main outcomes of the competition was that all the winning teams, including the GRIP team of the University Federico II, used some form of fusion of various forensic tools. In fact, only by using different and complementary techniques it is possible to detect a wide spectrum of forgeries, and to localize the tampered areas with good accuracy.

Motivated also by the experience gained in the IFS-TC Forensics Challenge, the work developed in this PhD thesis concerns different image detection and localization tools, making reference to different approaches, pixel-based, and camera-based. Therefore, the thesis is organized so as to devote a chapter to each one of these techniques. In more detail,

Chapter 1 presents the techniques based on the camera sensor noise (which can be considered as a sort of camera fingerprint), with special emphasis on a novel method proposed to improve the spatial resolution of the localization procedure. This method adopts a spatially adaptive filtering technique, the guided filter, with weights computed over the analyzed image. The experimental analysis shows that the proposed filtering strategy allows for a much better performance, especially in the critical case of small forgeries.

Chapter 2 faces the problem of detecting and localizing *copy-move forgeries*, where portions of the image are cut and pasted elsewhere in the same image to duplicate or hide objects of interest. We focus on dense-field techniques, which guarantee a superior performance with respect to their keypoint-based counterparts, usually at the price of a much higher processing time. The proposed technique overcomes the computational problem, thanks to several innovative solutions concerning all steps of the process. Experiments conducted on databases available online, prove the proposed technique to be at least as accurate, generally more robust, and typically much faster, than state-of-the-art dense-field references.

Chapter 3 addresses an innovative camera-based technique for tampering localization. This technique is based on *dense local descriptors* that are computed for each block of the image, and are eventually compared with a model to form a localization map of the forgery. Experiments show promising

results in many situations of interest, often superior to those of the much more complex camera-based techniques.

Chapter 1

PRNU-based localization of small-size forgeries

In this chapter we will focus on techniques based on the photo-response non-uniformity (PRNU) noise. These methods guarantee a good forgery detection performance irrespective of the specific type of forgery, since they do not detect the inserted object but rather the absence of the camera PRNU, a sort of camera fingerprint, dealing successfully with forgeries that elude most other detection strategies. The presence or absence of the camera PRNU pattern is detected by a correlation test. Given the very low power of the PRNU signal, however, the correlation must be averaged over a pretty large window, reducing the algorithm's ability to reveal small forgeries. To improve resolution, the correlation is estimated with a spatially adaptive filtering technique, with weights computed over the image. Experiments prove that this strategy allows for a much better detection performance in the case of small forgeries.

1.1 Introduction

Camera-based techniques discover image manipulations through peculiar traces left by the camera during the acquisition of the picture. One of the most promising to date relies on the photo response non-uniformity (PRNU) noise. The PRNU arises from tiny imperfections in the silicon wafer used to manufacture the imaging sensor [57]. These physical differences generate a unique sensor pattern, specific of each individual camera, constant in time, and independent of the scene, which can be therefore considered as a sort of camera fingerprint and used as such to accomplish forgery detection or camera

identification tasks. All the different types of tampering (copy-move, splicing, retouching) remove the original PRNU from the target area, enabling the detection of the forgery irrespective of the type of attack. PRNU-based techniques have proven quite robust to several forms of image processing [20, 27], including rotation, rescaling, and JPEG compression at relatively low rates (*e.g.,* Q=75). Given these precious properties, an intense research activity began on this topic as soon as the potential of the approach was recognized.

The first PRNU-based technique was proposed [77] in 2006. Blocks extracted from the estimated PRNU of the target image are compared with homologous blocks of the camera PRNU (estimated in advance from a set of sample images) and a tampering is declared whenever the normalized correlation falls below a given threshold. However, since the PRNU is a very weak signal, estimated by means of imperfect tools, its traces can be easily overwhelmed by noise in some regions of the image characterized by saturation or strong textures, leading to false alarms. Therefore, the same Authors propose in [20] a new version which reduces false alarms by identifying the potentially troublesome regions (through a predictor) and declaring them as genuine irrespective of the observed correlation index. Similar considerations guide the algorithm proposed in [75], where only regions with high signal quality are used, discarding those heavily deteriorated by irrelevant noise. In [68] a strategy to reduce the interference of scene details on the PRNU is proposed, while in [43] the suppression of non-unique artifacts is considered. These include, for example, JPEG block artifacts, and CFA interpolation artifacts, both characterized by regular "linear" spatially periodic patterns, relatively easy to correct [69]. Non-unique artifacts may lead to wrong results, especially in camera identification [48], because of the increased similarity between the PRNU fingerprints of a different devices with similar characteristics. In [108], canonical correlation analysis is used to increase the reliability of the decision variables. A better method for PRNU estimation based on nonlocal filtering is proposed in [24] and more recently in [26, 27] there is the reformulation of PRNU-based forgery detection as a Bayesian estimation problem.

The PRNU pattern is a very weak signal, it can be reliably detected only by jointly processing a large number of image samples, through a sliding-window analysis. The size of the sliding-window dictates therefore the effective resolution of the algorithm, causing forgeries smaller than the analysis window to remain often undetected. In [25], the authors resorted to a preliminary image segmentation to adapt the analysis window to the shape of candidate forgeries. Segmentation, however, is itself a source of errors, and the experimental anal-

ysis proved the heavy impact of such errors on performance. For this reason, the approach proposed in [23] replaces hard segmentation with a more flexible soft-segmentation strategy. In this chapter we will study in deep this approach that uses adaptive weights in the analysis window, computed on the basis of image content. A fast and effective implementation of this concept is obtained by resorting to guided filters [55]. Experiments prove that this algorithm provides much better results on critical small-size forgeries, with a negligible increase in complexity.

1.2 Background

In this section, the basic algorithm proposed in [77, 20] is described. Let $y \in \mathbb{R}^N$ be a digital image observed at the camera output, where y_i indicates the value at site i, either as a single color band or the composition of multiple color bands. Let us assume, in a simplified model [20, 57], that y can be written as

$$y_i = (1 + k_i)x_i + \theta_i = x_i k_i + x_i + \theta_i \tag{1.1}$$

where x is the ideal noise-free image, k the camera PRNU, and θ an additive noise term which accounts for all types of disturbances. The PRNU k is by now our signal of interest, very weak w.r.t. both additive noise θ and ideal image x. To increase the signal-to-noise ratio, we subtract from (1.1) an estimate of the ideal image, $\widehat{x} = f(y)$, obtained by means of a denoising algorithm, obtaining the so-called noise residual

$$\begin{aligned} r_i &= y_i - \widehat{x}_i = y_i k_i + (x_i - y_i)k_i + (x_i - \widehat{x}_i) + \theta_i \\ &= y_i k_i + n_i \end{aligned} \tag{1.2}$$

where, for convenience, k multiplies the observed image y rather than the unknown origina, and all disturbances have been collected in a single noise term n.

When a section of the image is tampered with, for example by replacing it with material drawn from other regions, the PRNU term is cancelled. Therefore, to decide about a possible forgery, PRNU-based techniques try to discover whether the PRNU term is present or not. In the following we briefly describe the technique proposed by Chen *et al.* [20], based on sliding-window analysis.

As a preliminary step, the true camera PRNU pattern, k, must be reliably estimated, which requires that either the target camera, or a large number of

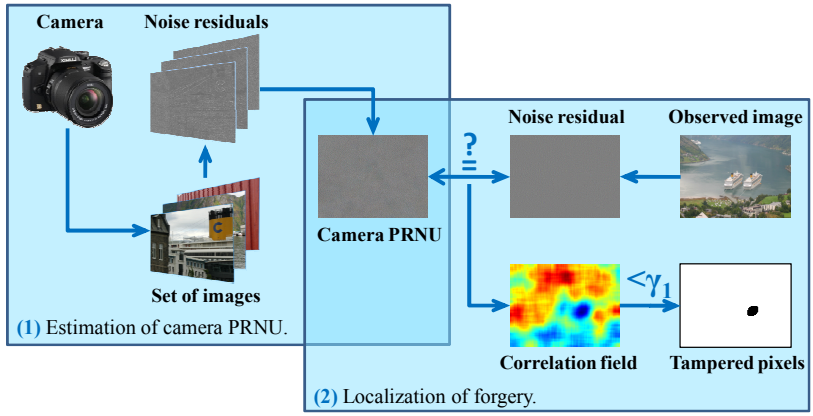


Figure 1.1: Scheme of PRNU-based forgery localization techniques.

photos taken by it, are available. Of course, this hypothesis is not always satisfied in real-world situations, representing the main limitation of this approach. The maximum likelihood estimate of the PRNU from M given images is computed in [20] as

$$\widehat{k}_i = \sum_{m=1}^{M} y_{m,i} r_{m,i} / \sum_{m=1}^{M} y_{m,i}^2 \tag{1.3}$$

where the weighting terms y_m account for the fact that dark areas of the image present an attenuated PRNU and hence should contribute less to the overall estimate. In the following, for the sake of simplicity, we will neglect the estimation error and will assume to know the camera PRNU perfectly, that is $\widehat{k} = k$.

Given k, the detection problem can be formulated as a binary test between hypothesis H_0 that the camera PRNU is absent (i.e. the pixel has been tampered with) and hypothesis H_1 that the PRNU is present (i.e. the pixel is genuine):

$$\begin{cases} H_0: & r_i = n_i \\ H_1: & r_i = z_i + n_i \end{cases} \tag{1.4}$$

with $z_i = y_i k_i$. The decision is based on the normalized correlation between r_{W_i} and z_{W_i}, namely, the restrictions of r and z, respectively, to a window W_i

centered on the target pixel:

$$\rho_i = \mathrm{corr}\left(r_{W_i}, z_{W_i}\right) = \frac{(r_{W_i} - \overline{r}_{W_i}) \odot (z_{W_i} - \overline{z}_{W_i})}{\|r_{W_i} - \overline{r}_{W_i}\| \cdot \|z_{W_i} - \overline{z}_{W_i}\|} \tag{1.5}$$

where $\odot$ denotes inner product, and $\overline{x}$ indicates mean of x. The algorithm proposed in [77] then compares the correlation with a threshold γ_1

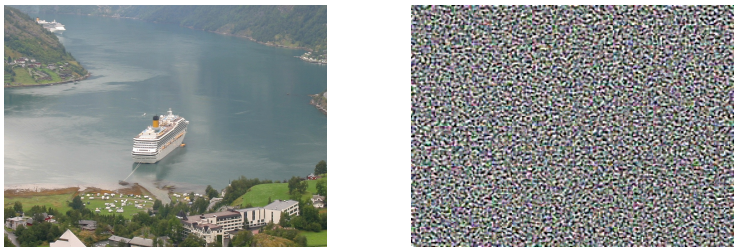
$$\widehat{u}_i = \begin{cases} 0 & \rho_i < \gamma_1 \\ 1 & \text{otherwise} \end{cases} \tag{1.6}$$

where $\widehat{u}_i \in \{0, 1\}$ is the algorithm output, 0 for forgery and 1 for genuine pixel. The threshold is selected according to the Neyman-Pearson criterion so as to guarantee a suitably small false acceptance rate (FAR) $\Pr(\widehat{u}_i = 1 \mid u_i = 0)$, with $u_i \in \{0, 1\}$ the true pixel class. Once fixed the FAR, however, there is no guarantee that the other type of error, the false rejection rate (FRR), remain reasonably small. In fact, under hypothesis H_1, the decision statistic is influenced by the image content. Even in the absence of forgery, the correlation might happen to be very low when the image is dark (since y multiplies the PRNU), saturated (because of intensity clipping), or when denoising does not perform well and some image content leaks into the noise residual. In figure 1.2(c) there is a correlation field computed on a genuine image, we can see that the correlation is lower in the textured regions when denoising did not work well. In [20] this problem is addresses by means of a "predictor" which, based on local images features, such as texture, flatness and intensity, computes the expected value $\widehat{\rho}_i$ of the correlation index under hypothesis H_1, an example of prediction is in figure 1.2(d). When $\widehat{\rho}_i$ is too low, indicating that no reliable decision can be made, the pixel is always labeled as genuine, the less risky decision, irrespective of the value of ρ_i. Therefore, the test becomes

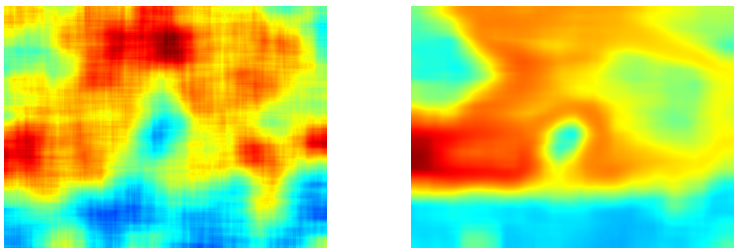
$$\widehat{u}_i = \begin{cases} 0 & \rho_i < \gamma_1 \ \text{ AND } \ \widehat{\rho}_i > \gamma_2 \\ 1 & \text{otherwise} \end{cases} \tag{1.7}$$

with γ_2 chosen heuristically by the user. Better strategies are considered in [26] and [27] where decisions are made jointly on all pixels based on a Bayesian/MRF modeling.

Although the above description remains necessarily at a conceptual level, it is worth going into some more detail for what concerns the decision statistic of equation (1.5). Given the low, and spatially varying, signal-to-noise ratio characterizing this problem, the two conditional pdf's $p_{\rho|H_0}(\cdot)$ and $p_{\rho|H_1}(\cdot)$



(a) Pristine image (b) Camera PRNU



(c) Correlation field (d) Prediction of correlation

Figure 1.2: Correlation field and its prediction for pristine image.

can overlap significantly, causing large probabilities of error. To obtain a reasonable separation between them, one is forced to compute the correlation over a large window, for example, 128×128 pixels, as done in [20]. By so doing, however, one is implicitly renouncing to detect forgeries much smaller than the window size (or just much thinner). In these cases, in fact, the analysis window comprises pixels of both types, forged and genuine, providing a highly unreliable decision statistic. In the original algorithm, in fact, detected forged regions smaller than 64×64 pixels (one fourth of the window size) are canceled right away, as they are more easily generated by random errors than by actual forgeries. Low resolution is therefore a major problem of this algorithm.

1.3 Methodology

To gain a better insight into our estimation problem let us elaborate some more on equation (1.5) introducing some simplifications. First of all, we neglect the means (which are typically negligible), and then focus only on the scalar product on the numerator, considering that the terms at the denominator serve only to normalize the correlation. Remember that $z = yk$ is the camera PRNU

multiplied point-wise by the input image and, likewise, $r = hy+n$ is the noise residual, with h the observed PRNU which might or might not coincide with k. Therefore, if we divide all terms point-wise by y, we obtain eventually the quantity

$$\tau_i = \frac{1}{|W_i|} \sum_{j \in W_i} \frac{r_j}{y_j} \frac{z_j}{y_j} = \frac{1}{|W_i|} \sum_{j \in W_i} (h_j + \frac{n_j}{y_j}) k_j \tag{1.8}$$

By defining a new noise field $\eta = nk/y$, and introducing generic weights ω_{ij}, eq.(1.8) becomes

$$\tau_i = \sum_{j \in W_i} \omega_{ij} (h_j k_j + \eta_j) \tag{1.9}$$

which can be interpreted as the linear filtering of the image hk affected by the additive noise η. In [20] the weights are all equal hence, a simple boxcar filtering is carried out.

Assuming that the whole analysis window is homogeneous, either genuine ($h = k$) or forged ($h \neq k$) and, for the sake of simplicity, that y is constant over the window, so that $\mathrm{VAR}[\eta_i] = \sigma_\eta^2$, we can characterize the random variable τ

$$E[\tau] = \begin{cases} \langle k^2 \rangle_i & h = k \\ 0 & h \neq k \end{cases} \tag{1.10}$$

$$\mathrm{VAR}[\tau] = \sigma_\eta^2 \sum_j \omega_{ij}^2 \tag{1.11}$$

where $\langle k^2 \rangle$ is the power of the camera PRNU estimated over W_i. In this condition, using uniform weights $\omega_{ij} = 1/|W_i|$ is indeed optimal, as it minimizes the variance of the estimate, and maximizes the probability of deciding correctly. However, if the analysis window is heterogeneous, that is, part of the pixels are genuine and part forged, the estimate will suffer a systematic bias, namely, the means will not be 0 or $\langle k^2 \rangle$ anymore, but some intermediate values, with an heavy impact on the decision reliability. In this case, the uniform weights are no more optimal, in general, and one should instead reduce the influence of pixels non homogeneous with the target by associating a small or even null weight with them.

This is exactly the problem arising in the case of small-size forgeries. By using a large analysis window with fixed weights we happen to include pixels of different nature, and the decision variable becomes strongly biased and basically useless, even in favourable (bright, smooth, unsaturated) areas of the

image. If we could find and include in the estimation only predictors homogeneous with the target, all biases would disappear, although at the cost of an increased estimation variance.

The bias / variance trade-off is indeed well-known in the denoising literature. This problem has received a great deal of attention, recently, in the context of nonlocal filtering [37, 38, 33], the current state of the art in denoising, where predictor pixels are weighted based on their expected similarity with the target. The similarity, in its turn, is estimated by comparing patches of pixels centered, respectively, on the target and on each candidate predictor pixel: when the patch surrounding a predictor is similar to the target patch, the predictor is assumed to be similar to the target, and a large weight is associated with it. This approach cannot work as is with our noise-like input image, rz, as it lacks the geometrical structures that help computing a meaningful similarity measure. However, we can take advantage of the original observed image y, using it as a "pilot" (again a well-known concept in denoising) to compute similarities, and applying the resulting weights in the actual filtering of the rz field.

Unfortunately, nonlocal filtering, with its intensive patch-based processing, is characterized by high computational complexity, which becomes unacceptable in our case, where the weak PRNU signal calls for large filtering windows. We resort therefore to a different implementation of this basic idea, based on guided filtering, a recently proposed [55] technique which implements nonlocal filtering concepts by leveraging heavily on the use of a pilot image associated with the target image (see Fig.1.3).

We recall here the basics of guided filtering following closely the development and notation used in [55], and referring the reader to the original paper for a more detailed treatment. Let p be the image to be filtered, q the filter output, and I a pilot image assumed to bear valuable information on p. We consider linear filtering, in the form

$$q_i = \sum_j \omega_{ij} p_j \tag{1.12}$$

Then, we assume that, locally to each pixel i, q depends linearly on I, that is

$$q_j = a_i I_j + b_i, \quad \forall j \in \Omega_i \tag{1.13}$$

where Ω_i is a square window of radius R centered on i. The parameters a_i and b_i are chosen to minimize over Ω_i the squared error between observed image

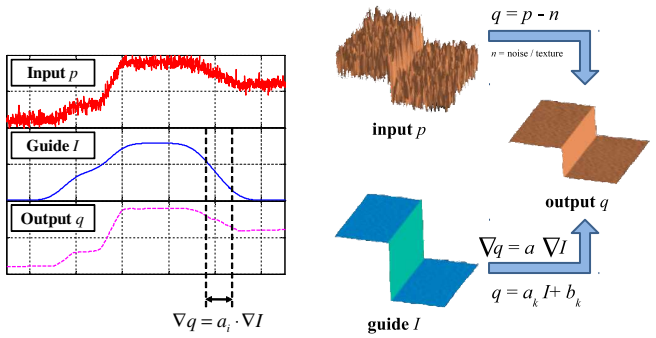


Figure 1.3: Scheme of guided filtering [55].

and model

$$(a_i, b_i) = \arg\min_{(a,b)} \sum_{j \in \Omega_i} [(a_i I_j + b_i - p_j)^2 + \varepsilon a_i^2] \tag{1.14}$$

with ε a regularizing parameter that penalizes large values of a. The optimal values are

$$a_i = \frac{1}{|\Omega_i|} \sum_{j \in \Omega_i} \frac{I_j\, p_j - \bar{I}_i\, \bar{p}_i}{\sigma_i^2 + \varepsilon} \tag{1.15}$$

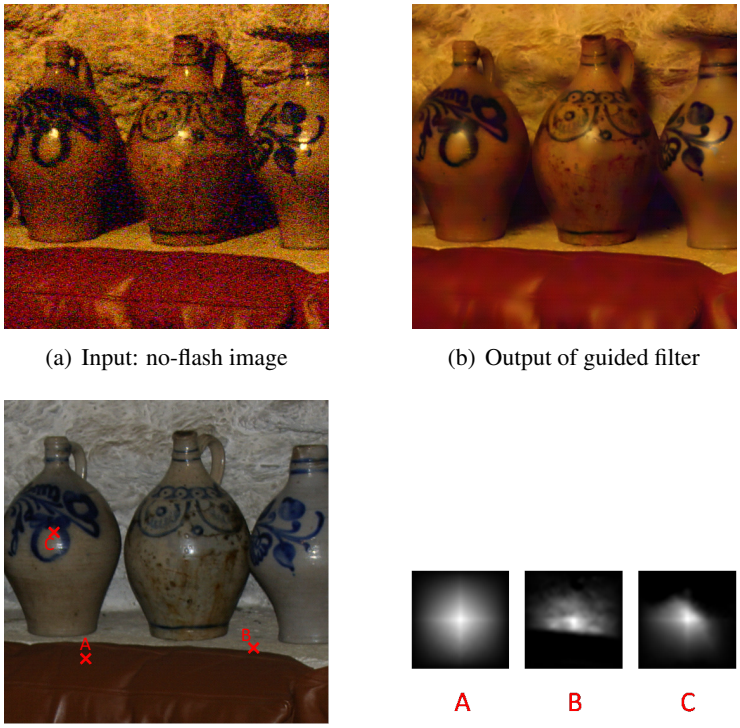
$$b_i = \bar{p}_i - a_i \bar{I}_i \tag{1.16}$$

where $\bar{p}_i$ and $\bar{I}_i$ indicate the average of p and I over Ω_i, and σ_i^2 is the variance of I over Ω_i. By substituting the optimal values back into (1.13) we obtain an estimate of q_j for all output pixels in the window Ω_i. Each of these pixels, however, falls in several such windows, and hence, to obtain the final filtered value, we average all such estimates

$$q_j = \frac{1}{|\Omega_j|} \sum_{i \in \Omega_j} (a_i I_j + b_i) = \bar{a}\, I_j + \bar{b} \tag{1.17}$$

which is the final expression of the linear filtering process of p guided by the pilot image I under the local linear model (1.13).

Fig.1.4 shows an example of application of guided filtering where a picture taken without flash is denoised using its flash version as guidance. The main reason for reporting above all the intermediate expressions is to point out that all the computation amounts to a few boxcar filtering, applied to p, I, I^2, a, and b, and carried out by integral image techniques with negligible complexity.



(a) Input: no-flash image

(b) Output of guided filter

(c) Guide: flash image

(d) Masks: weights adapt to the local guide content

Figure 1.4: An application of guided filtering to denoise a no-flash image under the guidance of its flash version. [55]

For our algorithm [23], of course, the input image is the product rz, the output is the decision statistic ρ, while the pilot (scalar) image can be a combination of the color bands of the original image, y or its denoised version x, or any suitable field of features extracted from these images. By tuning the two parameters of the filter, the window radius R and the regularizing parameter ϵ, the influence of the pilot image in the filtering process can be modulated at will.

1.4 Experimental evaluations

To prove the potential of the proposed approach we begin by showing, in Fig.1.6, a few sample images and the corresponding correlation fields. The

image on the first row presents a large forgery, easily detectable in both the correlation fields (last two columns) as the region is much darker than in the predicted field (middle column). On the second and third row, instead, we have quite small forgeries, which leave little or no trace in the field computed by boxcar filtering, while are clearly detectable in the field obtained by guided filtering. Although these last examples are very favourable for the guided filtering approach, due to the high contrast between forgeries and background, they make clear that the original image can help making a better decision.

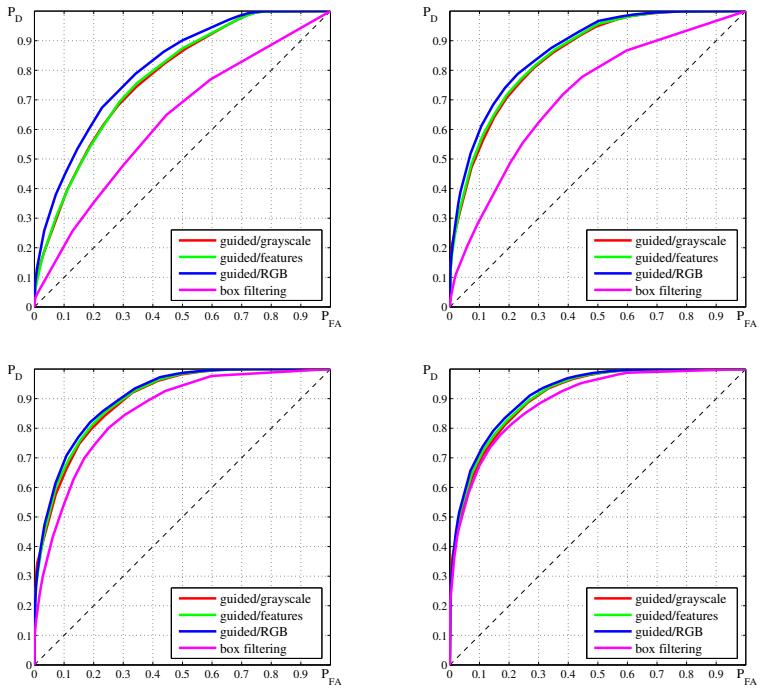


Figure 1.5: ROCs obtained with boxcar and guided filtering with forgeries of size: 48×48, 64×64, 96×96, and 128×128 pixels.

A more extensive experimental analysis is presented in Fig.1.5 where we show the receiver operating curves (ROC) obtained using the original boxcar filtering and the proposed guided filtering. To ensure a fair comparison, the algorithm proposed in [20] is used in all cases, with its wavelet-based denoising filter [80] and the two-threshold test, and we change only the way the correlation field is computed. In particular, for guided filtering we consider three implementations, using as pilots, respectively, *i)* the grayscale version of the

original image y, *ii)* the RGB version of the same image, and *iii)* the vectorial image composed by the four features [20] used to design the correlation predictor. We use a test set of 200 uncompressed 768×1024-pixel images with a square forgery at the center, drawn at random from a different image. The camera (a Canon EOS-450D) PRNU is estimated off-line on a separate training set, used also to design the predictor. Each ROC is the upper envelope of pixel-level (P_D, P_{FA}) points obtained as the algorithm parameters vary. For guided filtering we used ε =0.16 and R =32, which corresponds to an analysis window of 128×128. This window size is also used for boxcar filtering, and in all cases, to allow a fair comparison, the minimum size of acceptable detected forgeries was lowered to 32×32 pixels. Comparison is carried out separately for very-small, small, medium and large forgeries. With forgeries of dimension 48×48 pixels and 64×64 pixels (first two graphs), guided filtering guarantees a large performance improvement over boxcar filtering, synthesized by the area under curve (AUC) figure which grows from 0.63 to 0.78 in the first case and from 0.71 to over 0.85 in the second. With medium-size forgeries, 96×96 pixels, the performance gain is much more limited, with the AUC growing from 0.85 to 0.90, and becomes almost negligible, as expected, with larger 128×128 forgeries. No significant difference is observed, instead, as the pilot image changes, with the RGB pilot only slightly preferable to the others.

1.5 Conclusions

We proposed a new strategy to improve the resolution of PRNU-based forgery detection techniques. The basic idea is to exploit the image structure to better estimate the correlation field on which decisions are based. This is obtained here by resorting to the guided filtering approach, obtaining a very fast algorithm, characterized by a performance much superior to the reference technique when small forgeries are involved. In the ongoing research, we are experimenting with other pilot images, studying in more depth the dependance on the algorithm parameters, and assessing performance in a wide variety of conditions, including various forms of distortion, and JPEG compression.

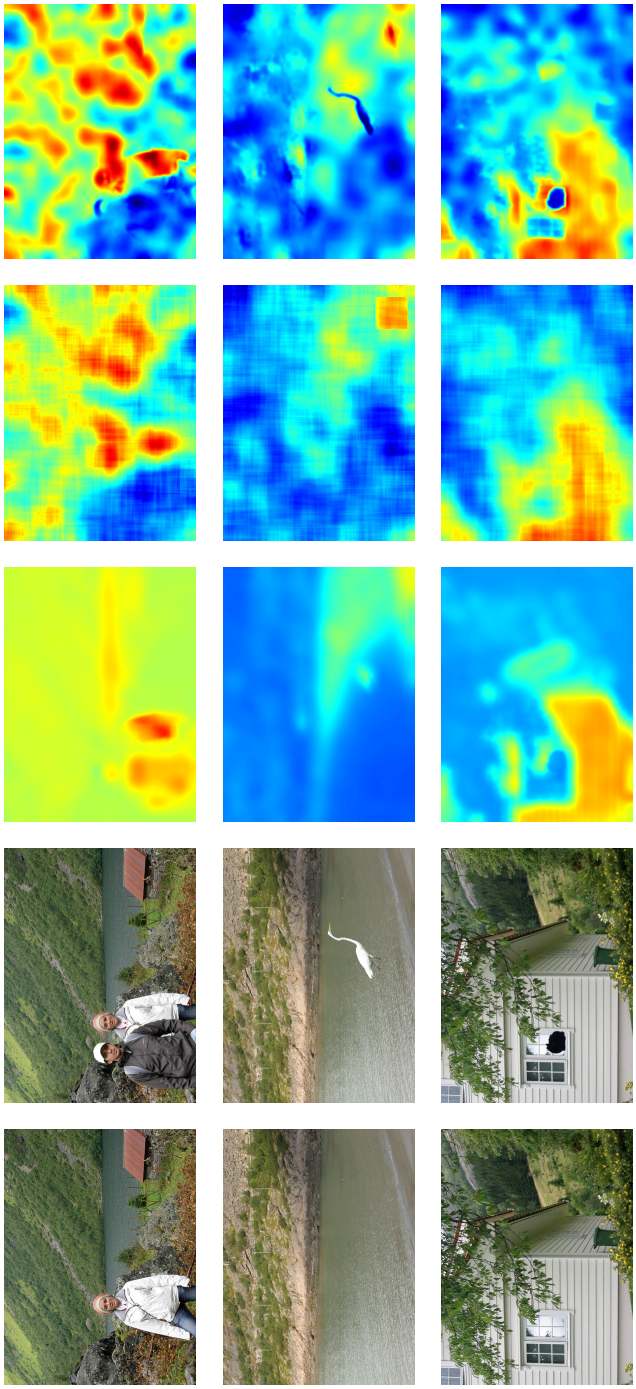


Figure 1.6: Sample results. From left to right, original and forged image, correlation field predicted, and computed by boxcar and guided filtering.

Chapter 2

Efficient dense-field copy-move forgery detection

A very common type of manipulation is the *copy-move forgery*. In this case one or more regions of an image are cut and pasted elsewhere, in the same image, in order to duplicate or hide objects of interest. In fig.2.1, there are two examples of copy-move forgeries. In the first case (top row), an object has been duplicated. In the second example, a smooth region has been duplicated to cover an undesired detail. In fact, these forgeries are extremely simple to perform. In particular the forged images in fig.2.1 have been made through the clone stamp tool availble in modern image editing tools, such as Photoshop or Gimp. Detecting these forgeries, however, can be challenging, especially in the occlusive case, where pieces of smooth background are duplicated. Indeed, in the last years many papers have dealt with this problem, as reported in the review paper [5]. In this chapter, we present a new algorithm for the accurate detection and localization of copy-move forgeries, based on rotation-invariant features computed densely on the image [35, 36]. This proposal, tested on databases available online, is very efficient and accurate.

2.1 Related work

All detection algorithms proposed to detect copy-move forgeries follow a common pipeline [29] based on three steps

- *feature extraction:* a suitable feature is computed for each pixel of interest, expressing the image behavior in its neighborhood;

Figure 2.1: Examples of copy-move forgeries. The genuine images on the left and the forged images on the right.

- *matching:* the best matching of each pixel is computed, based on the associated feature;
- *post-processing:* the offset field, linking pixels with their nearest neighbors, is filtered and processed in order to reduce false alarms.

These operations can be carried out for each pixel of the image, generating thus a dense offset field, or for just some selected keypoints, in which case the field is sparse. Keypoint-based methods, working on a relatively small set of pixels, are usually much faster than those based on dense matching. Therefore, they can afford to compute long and complex features to associate with the keypoints, characterized by rotation/scale invariance, like SIFT [87, 7], SURF [95], LBP [110], DAISY [53], or even robust to some geometric transformations, like in [62, 19]. Unfortunately, they are intrinsically less accurate than dense-field methods, especially when copy-moves involve only smooth regions, which is typically the case with occlusive forgeries. This performance gap, which appears clearly in the benchmarking paper [29], is the main reason driving us to focus on the dense-field approach.

The obvious problem, in this case, is complexity, since all pixels require the three phases of feature extraction, matching, and post-processing. Therefore, feature extraction is required to be intrinsically simple, and to produce

features as short as possible to speed-up the matching phase. Using RGB values is a possible choice [67], but the resulting features tend to be unnecessarily long, and performance may be severely affected by JPEG compression, noise addition, and other common distortions. In the literature, to improve robustness, features are typically extracted through some transforms, like DCT [45, 104, 103], Wavelet [83], PCA [78], SVD [109], reducing also their length, thanks to the decorrelation of coefficients. Such features, however, do not perform well in the presence of rescaling and rotation. Therefore, a significant effort has been devoted, recently, towards the definition of features that deal satisfactorily with these situations. Circular harmonic transforms are well-suited to provide rotation-invariance, and several possibilities have been tested to this end, including Zernike moments [91], and polar sine and cosine transforms [71, 70]. As for scale-invariance, research has mostly focused on variations of the Fourier-Mellin Transform [11, 59, 100, 101], based on a log-polar sampling. The same sampling scheme is also carried out directly in the spatial domain in [16], producing a simple one-dimensional descriptor.

Besides feature selection, in the literature much attention has been devoted to the matching phase itself. In fact, plain exhaustive search is prohibitive due to its huge complexity, and faster techniques must be devised to produce the offset field in a reasonable time. A significant speed-up can be obtained only renouncing exact matching, and adopting some approximate nearest-neighbor search strategy, in which case accuracy becomes a further aspect of interest. Some techniques [11, 16] rely on simple lexicographic sorting, but this approach is very sensitive to noise and other forms of impairment. Robustness improves significantly with more sophisticated fast search techniques, like kd-trees [84], used in [67, 29], or locality sensitive hashing [47], considered in [91, 71]. These state-of-the-art matching techniques, however, are designed to work for very generic problems, like the retrieval of documents over the Internet. Therefore, they do not take into account a major circumstance of interest in this context, namely, that the we are looking for a nearest-neighbor *field*, where features are extracted from a real-world image, and are not just a collection of unrelated queries. This observation is at the core of [36, 35] in which we propose an efficient and accurate technique for copy-move detection and localization, which deals successfully also with a number of geometric transformations.

2.2 Proposed method

Our method, proposed in [36], follows the general workflow considered in [29], but proposes innovative and efficient solutions for most of the key steps of the workflow. Matching of dense features, in particular, is carried out through the PatchMatch algorithm [9], specific for nearest-neighbor search over images, which greatly reduces the processing time while providing an accurate and regular offset field. Efficiency is pursued also in the other processing steps, by selecting compact, low-complexity features, and by implementing a simple and reliable post-processing scheme, which fully exploits the smoothness of PatchMatch offset field. As a result, our dense-field technique turns out to be nearly as fast as keypoint-based techniques, but much more reliable than them. Moreover, by selecting suitable scale and/or rotation-invariant features, higher robustness w.r.t. a number of geometric distortions is also achieved. With reference to the general scheme recalled in Section 2.1, we will first focus on efficient matching, and post-processing; then we will consider and discuss several features with scale-invariance and rotation-invariance properties.

2.2.1 Computing a dense NNF

Let

$$I = \{I(s) \in R^K, s \in \Omega\} \tag{2.1}$$

be an image defined over a regular rectangular grid Ω. With each pixel[1], s, we associate a feature vector, $f(s)$, which describes the P-pixel image patch centered on s. In the simplest case, $f(s)$ might just be the KP-vector formed by stacking all image values observed in the patch. More often, to improve efficiency, the feature is a compact description of the patch, with length much smaller than KP.

Given a suitable measure of distance between features, $D(f', f'')$, we define the nearest neighbor of s as the pixel, $s' \in \Omega, s' \neq s$, which minimizes this distance over the whole image. Rather than the nearest neighbor field (NNF) itself, in the following we will often consider the equivalent offset field, $\{\delta(s), s \in \Omega\}$, where

$$\delta(s) = \arg \min_{\phi: s+\phi \in \Omega, \phi \neq 0} D(f(s), f(s+\phi)) \tag{2.2}$$

and, of course, the NN is $s' = s + \delta(s)$.

[1] To the extent possible, to keep a light notation, we avoid double indices and boldface, using the single normal-type variable s to indicate pixel location.

Finding the *exact* NNF is computationally very demanding, even for a relatively small image, since the complexity grows quadratically with the image size. Most real-world applications, however, do not really need the exact NN, and perform almost as well with some good approximation of it. Finding an approximate NN can be orders of magnitude less expensive than finding the exact one, with a speeding factor depending on feature statistics and on the desired accuracy. As a matter of fact, a large number of techniques have been proposed in recent years for this task, the most promising of which are based on kd-trees and on hashing. Indeed, some of these techniques have been already applied in the context of copy-move detection, as in [29], where feature matching is carried out through the kd-tree based methods of the FLANN package [84], or in [91], where locality-sensitive hashing [47, 8] is used.

These techniques, however, are not really suited for the problem under analysis, as they consider each feature to match as an *independent* query, neglecting altogether the spatial regularity typical of natural images. Image smoothness and self-similarity, instead, imply that the NNs of close pixels are very often spatially close themselves. By exploiting this simple property one can both reduce the computational cost and improve the quality of the final NNF. Indeed, the search can be accelerated by using the offsets of neighboring pixels as initial guesses for the current one. This approach is well-known in the denoising literature, where it is exploited for nonlocal filtering techniques based on block-matching. Moreover, by resorting to spatial prediction, smoother offset fields are automatically obtained. This is extremely valuable in the context of copy move detection, as it allows one to avoid complex (and time-consuming) post-processing algorithms to regularize the field afterwards.

Based on these considerations, we resort here to a technique recently proposed for the specific problem of fast NNF computation over images.

A. The PatchMatch algorithm

PatchMatch [9] is a fast randomized algorithm which finds dense approximate nearest neighbor matches between image patches.

Initialization. The offset field is initialized at random, as

$$\delta(s) = U(s) - s \tag{2.3}$$

where $U(s)$ is a bi-dimensional random variable, uniform over the image support Ω. We note explicitly, here, that $\delta(s) = 0$ is obviously discarded, as it corresponds to a trivial and useless solution. Likewise, since we are looking

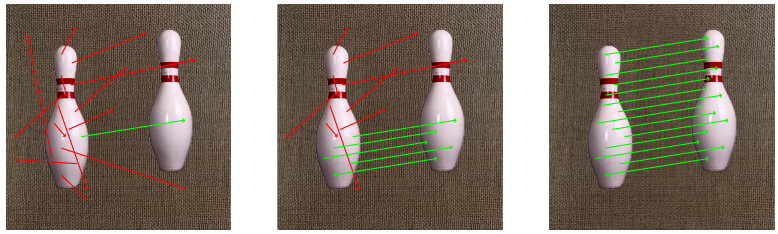


Figure 2.2: Offset propagation. Initially (left), most offsets are wrong (red); a random good offset (green) propagates along the scanning order (center) until the whole copy-moved region is covered (right).

for matches relatively far apart from the target, we exclude all offsets smaller than a given threshold, $\|\delta(s)\|_\infty < T_{D1}$, a condition applied implicitly in all further developments. Most of the initial random offsets are just useless, but it is very likely that a certain number of them will be optimal or near-optimal. The main idea of PatchMatch is to quickly propagate such good offsets, updating iteratively the whole field. In the generic iteration, there are two phases: propagation and random search.

Propagation. In this phase the image is raster scanned top-down and left-to-right, and for each pixel s the current offset is updated as

$$\delta(s) = \arg \min_{\phi \in \Delta^P(s)} D(f(s), f(s+\phi)) \tag{2.4}$$

where $\Delta^P(s) = \{\delta(s), \delta(s^r), \delta(s^c)\}$, and s^r and s^c are the pixels preceding s, in the scanning order, along rows and columns, respectively. In practice, the algorithm checks whether the offsets associated with the causal neighbors improve the matching quality w.r.t. the current one. Therefore, if a good offset is available for a given pixel of a region with constant offset, this will very quickly propagate, filling the whole region below and to the right of it. To avoid biases, the scanning order is then reversed (bottom-up and right-to-left) at every other iteration. In Fig.2.2, using a simple toy example, we provide some insight into the rationale of this procedure for the copy-move detection application. On the left, some of the initial offsets are shown (only on a bowling pin, to avoid cluttering the figure), which are all wrong (red) except for one of them (green), which is correct by chance. After the first iteration of the algorithm, the correct offset propagates to the bottom-right part of the pin (center), and to all of it after the second iteration (right). With more complex geometries, the complete propagation might require some more iterations.

Random search. The above propagation procedure is obviously greedy, and as such suboptimal, depending on the quality of the random initialization. Therefore, to minimize the risk of being trapped in local minima, after the updating of equation (2.4), a random search phase is also considered, based on a random sampling of the current offset field. The candidate offsets $\delta_i(s), i = 1, \ldots, L$ are chosen as

$$\delta_i(s) = \delta(s) + R_i \tag{2.5}$$

where R_i is a bi-dimensional random variable, uniform over a square grid of radius 2^{i-1}, excluding the origin. In practice, most of these new candidates are pretty close to $\delta(s)$, but large differences are also allowed, with small probability. The random-search updating reads therefore as

$$\delta(s) = \arg \min_{\phi \in \Delta^R(s)} D(f(s), f(s + \phi)) \tag{2.6}$$

where $\Delta^R(s) = \{\delta(s), \delta_1(s), \ldots, \delta_L(s)\}$.

For an image of, say, 1024×1024 pixels, $L \leq 10$. Considering that the procedure typically converges after a few iterations, the whole computational load is in the order of 10^2 feature distance computation per pixel, as opposed to 10^6 for full-search, which fully explains the algorithm speed. Of course, PatchMatch relies on the implicit hypothesis that the NNF is mostly regular, and in particular regular over the regions of interest where a match is looked for, otherwise the crucial propagation step would be basically ineffective. However, this is exactly the condition encountered in copy-move detection.

B. Modifying PatchMatch to deal with rotation and rescaling

The basic algorithm described above does not deal with scale changes and rotations, which are instead very common in copy-move image tampering. In a subsequent paper [10], however, the same authors of PatchMatch generalized and extended it under several respects, including the ability to search across scales and rotation angles, going beyond mere translations, and to match patches based on arbitrary descriptors and distances, rather than just the Euclidean norm of original patches used in the basic version.

The solution proposed in [10] is straightforward: rather than analyzing only the 2d space spanned by the offset components (δ_1, δ_2), a 4d space is considered, comprising two further dimensions, scale, α, and rotation θ. All steps are then carried out as before, with obvious adjustments of minor significance, except for two main differences:

1. given the current values of α and θ, the target patch is suitably rescaled/rotated, interpolated and resampled, to be comparable with the original one;

2. likewise, in the propagation step, the candidate offsets are not just copied from the neighbors' offsets, but computed based on them and on the local transformation identified by α and θ.

The generalized algorithm preserves thus the simplicity of the original version, an appealing property. However, it presents some significant drawbacks. First of all, the computational complexity increases sharply, not only for the need to carry out patch interpolation but also for the increased number of iterations necessary to converge in a 4d search space. Moreover, the algorithm is not amenable to be used with compact features, like those described in Section 2.2.3, renouncing their potential for higher descriptive power and scale/rotation invariance, as well as the reduced complexity associated with their shorter length. Last, but not least, experiments on copy-move forgery detection [35] show generalized PatchMatch to be much less reliable than the basic version in the fundamental case of simple rigid translation, causing a significant loss in performance. Our conjecture is that with an higher-dimensionality optimization space, a large number of suboptimal matchings are available, which trap the algorithm into local minima. This problem should be solved by the random search phase but random sampling becomes too sparse in such a large space, and hence less effective.

Given these dismaying results, in terms of both accuracy and complexity, in [35] we proposed a different modification of the basic PatchMatch algorithm, based on the use of scale/rotation invariant features. In fact, numerous such features have been proposed in recent years to describe image patches, solving in advance (as far as invariance holds) the problem of matching patches subject to such transformations.

Thanks to the use of invariant features, the basic algorithm needs be modified exclusively for what concerns the propagation phase, by changing the set of candidate offsets available for updating. In the original algorithm, the current offset for pixel s is compared with two other candidate offsets, $\delta(s^r)$ and $\delta(s^c)$, which are simply the causal zero-order predictions of $\delta(s)$ along image rows and columns, renamed here accordingly as $\widetilde{\delta}^{0r}(s)$ and $\widetilde{\delta}^{0c}(s)$, respectively. Of course, zero-order predictors are effective only in regions characterized by a constant offset, corresponding to copy-moves with rigid translations. Rotated, rescaled, and rotated-rescaled copy-moves, instead, are described by

linearly varying offset fields, as shown in Fig.2.3, in which case, first-order predictors can be expected to work correctly.

Therefore we enlarge the set of candidates in (2.4) considering both zero-order and first-order predictors

$$\begin{aligned} &\widetilde{\delta}^{0x}(s) = \delta(s^x) \\ &\widetilde{\delta}^{1x}(s) = 2\delta(s^x) - \delta(s^{xx}) \\ &x \in \{r, d, c, a\} \end{aligned} \tag{2.7}$$

where s^{xx} is the pixel preceding s^x along direction x in the scanning order (see Fig.2.4), and we include also the diagonal and antidiagonal directions, d and a, respectively, obtaining eventually the enlarged set of predicted offsets

$$\begin{aligned} \Delta^P(s) \quad = \quad & \{\delta(s), \widetilde{\delta}^{0r}(s), \widetilde{\delta}^{0d}(s), \widetilde{\delta}^{0c}(s), \widetilde{\delta}^{0a}(s), \\ & \widetilde{\delta}^{1r}(s), \widetilde{\delta}^{1d}(s), \widetilde{\delta}^{1c}(s), \widetilde{\delta}^{1a}(s)\} \end{aligned} \tag{2.8}$$

With this modification, whenever a correct offset field is found over a couple of neighboring pixels it will quickly propagate to the rest of the interested region within two iterations. Moreover, thanks to the zero-order predictor, and to the random search phase, it is not difficult to reach the initial condition which triggers the propagation.

Some comments are in order. First of all, the complexity of the proposed version is very close to that of the basic one, as only the propagation phase is modified, and in a quite inexpensive way. Moreover, the opportunity to adopt compact features in place of pixel values grants a stronger efficiency gain. As for accuracy, using features that are scale and rotation invariant, we can match a very general class of copy-moves. It could be pointed out that most features proposed in the literature are only rotation-invariant. However, PatchMatch

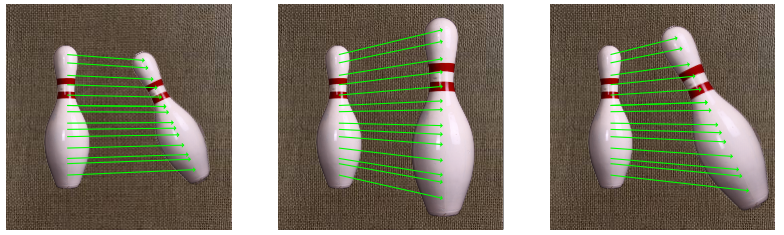


Figure 2.3: Offsets vary linearly over the copy-moved region in the presence of rotation (left), rescaling (center), or both (right).

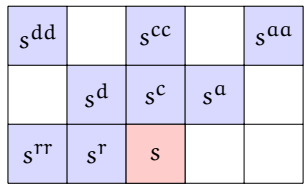


Figure 2.4: Predictor geometry for direct scanning. Blue pixels are used to build various zero- and first-order predictors for the offset of pixel s (red).

exhibits a remarkable robustness w.r.t. limited scale changes [35], thanks to the random search phase. Hence, one can explore the scale dimension through a brute-force approach in a limited number of steps, a non-elegant solution, viable only thanks to PatchMatch speed, which proved to be very effective in practical applications [31, 32]. Finally, it should be pointed out that, on the wake of PatchMatch, other dense-field techniques have been proposed recently [65, 86], providing further improvements in search efficiency by replacing the random search phase with some smarter initialization, based on fast approximate NN search techniques. This is a very active field of research, and we are confident that some of these ideas may be included in future versions of our algorithm.

2.2.2 Post-processing based on dense linear fitting

Ideally, the offset field obtained through feature matching should be mostly chaotic except for some large smooth regions with linear behavior in correspondence of cloned objects. In practice, because of noise, compression, geometric deformations, illumination changes, look-alike regions, the computed offset field rarely follows this model and some post-processing is necessary to

1. regularize the offset field to increase the probability of detecting actual copy-moves;

2. add some suitable constraints to reduce the probability of false alarms.

The first problem is especially challenging, and previous papers tackled it through sophisticated and relatively slow methods, such as the well-known RANSAC [91] or SATS [28]. In our case, however, thanks to the implicit filtering enacted by PatchMatch, the offset field is regular enough to consider a simpler approach, based on dense linear fitting (DLF).

We want to fit, in a suitable N-pixel neighborhood of s, the true offset field $\delta(s)$ through a linear (more precisely, affine) model

$$\widehat{\delta}(s_i) = As_i, \;\; i = 1, \ldots, N \tag{2.9}$$

with the parameters of the transformation, A, set so as to minimize the sum of squared errors w.r.t. the true data

$$\epsilon^2(s) = \sum_{i=1}^{N} \|\delta(s_i) - \widehat{\delta}(s_i)\|^2 \tag{2.10}$$

Although the offset field is bi-dimensional, the model parameters can be optimized independently for each of the two components, so in the following lines, in order to simplify notations, we will treat $\delta(s)$ as a single component field.

With this understanding, we can write the problem as

$$a^{\text{opt}} = \arg\min_{a} \|\delta - Sa\|^2 \tag{2.11}$$

where $\delta = [\delta(s_1), \delta(s_2), \ldots, \delta(s_N)]^T$ is the vector of offsets output by the matching phase, $a = [a_0, a_1, a_2]^T$ is the vector of parameters that identifies the affine transform, and S is the $N \times 3$ matrix of the homogeneous coordinates of all pixels in the neighborhood

$$S = \begin{bmatrix} 1 & s_{11} & s_{12} \\ 1 & s_{21} & s_{22} \\ & \vdots & \\ 1 & s_{N1} & s_{N2} \end{bmatrix} \tag{2.12}$$

so that,

$$\widehat{\delta}(s_i) = a_0 + a_1 s_{i1} + a_2 s_{i2}, \;\; i = 1, \ldots, N \tag{2.13}$$

This is a well known multiple linear regression problem [66], with solution

$$a^{\text{opt}} = (S^T S)^{-1} S^T \delta \tag{2.14}$$

The corresponding sum of squared errors (SSE) is therefore

$$\begin{aligned} \epsilon^2(s) &= \|\delta - S(S^T S)^{-1} S^T \delta\|^2 \\ &= \|(I - H)\delta\|^2 \\ &= \delta^T (I - H)\delta \end{aligned} \tag{2.15}$$

where we have exploited the fact that $H = S(S^T S)^{-1} S^T$ is symmetric and idempotent ($HH = H$). If the coordinates in (2.12) are taken relative to s, and the neighborhood has constant shape, the matrix H does not depend on s, hence computing the SSE reduces to evaluating the quadratic form (2.15) for the two offset components. However, the processing cost can be further reduced by decomposing the rank-3 matrix H as

$$H = QQ^T, \qquad Q = [q_1, q_2, q_3] \tag{2.16}$$

where q_j is a column vector of length N, and hence

$$\epsilon^2(s) = (\delta^T \delta) - (\delta^T q_1)^2 - (\delta^T q_2)^2 - (\delta^T q_3)^2 \tag{2.17}$$

computed through a few filtering operations and some products.

We can now outline the complete post-processing procedure, which comprises the following steps:

1. median filtering on a circular window of radius ρ_M;
2. computation of the fitting error, $\epsilon^2(s)$, w.r.t. a least-squares linear model over a circular neighborhood of radius ρ_N;
3. thresholding of $\epsilon^2(s)$ at level T_ϵ^2;
4. removal of couples of regions closer than T_{D2} pixels;
5. removal of regions smaller than T_S pixels;
6. mirroring of detected regions;
7. morphological dilation with a circular structuring element of radius $\rho_D = \rho_M + \rho_N$.

As already said in Section 2.2.1, a locally linear model is certainly appropriate for the copy-moves considered in this chapter, but minimum mean-square error (MMSE) fitting is very sensitive to outliers. Therefore, before the DLF, we carry out a median filtering process, which removes outliers, but leaves the signal unaltered where it has a linear behavior. In step 3, the image is segmented to single out candidate copy-moved regions. Here, to keep complexity limited, a simple thresholding is considered, but plenty of methods are available to improve this process. Steps 4 and 5 are meant to remove matchings obtained by chance. Spurious matchings abound in natural images

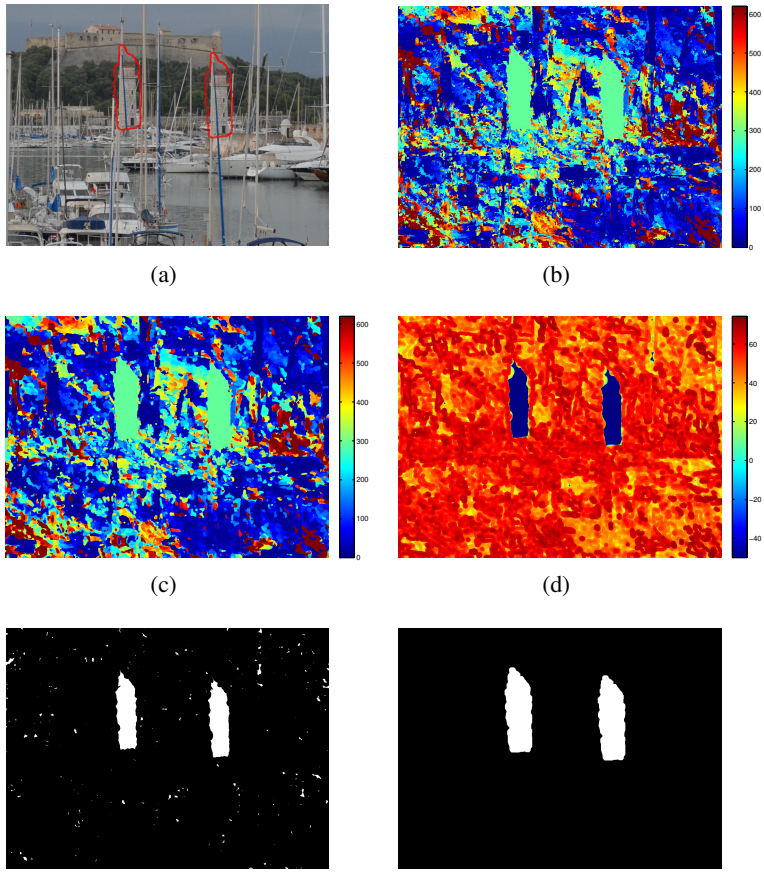
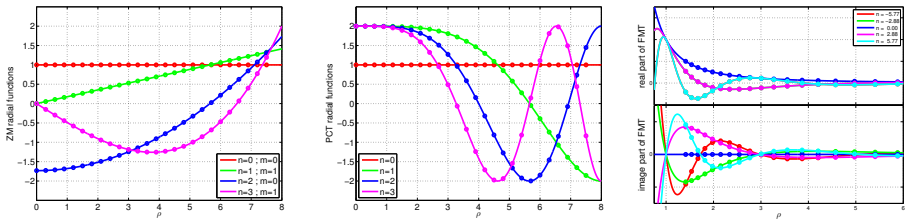


Figure 2.5: Post-processing steps: (a) original forged image, (b) magnitude of offsets, (c) median filtering, (d) fitting error $\epsilon^2(s)$ (dB), (e) thresholding of $\epsilon^2(s)$, (f) final mask.



(a) (b) (c)

Figure 2.6: Radial profiles of some CHTs: Zernike (left), PCT (center), FMT (right).

because of repeated patterns or uniform background. In the first case, however, similar details are typically quite small, and can be removed based on the size constraint. In the second case, we exploit the fact that background regions are generally not as uniform as they appear. Gradual luminance changes imply that, in a uniform background region, patches similar to the target are also close to it, and hence the additional constraint on distance eliminates these regions. It goes by itself that an image portraying multiple replicas of the same object can still give rise to false alarms, especially if scale and rotation invariant features are used. Once we decide that pixel s belongs to a copy-move region, it make sense to mark as copy-moved also pixel $s + \delta(s)$, which we do in step 6. Finally, considering that both median filtering and model fitting tend to erode the support of the copy-moved regions, in step 7 we restore them through a complementary dilation.

The various steps of the procedure are illustrated In Fig.2.5 on a real-world example.

2.2.3 Feature Extraction

A large number of features have been proposed in recent years for the purpose of copy-move detection, and many of them have been considered in the extensive experimental comparison carried out in [29]. Here, we will focus on features based on the family of Circular Harmonic Transforms (CHT) [58] which possess desirable invariance properties.

Let $I(x, y)$ be a scalar image defined on a continuous space, $(x, y) \in R^2$, and let $I(\rho, \theta)$ be its representation in polar coordinates, with $\rho \in [0, \infty]$ and $\theta \in [0, 2\pi]$. The CHT coefficients are evaluated by projecting the image over

the basis functions $K_{n,m}(\rho,\theta)$ of the transform

$$F_I(n,m) = \int_0^{2\pi}\int_0^{\infty} I(\rho,\theta)\, K^*_{n,m}(\rho,\theta)\rho\, d\rho\, d\theta \tag{2.18}$$

The basis functions have the form

$$K_{n,m}(\rho,\theta) = R_{n,m}(\rho)\frac{1}{\sqrt{2\pi}}e^{jm\theta} \tag{2.19}$$

that is, they are obtained as the product of a radial profile $R_{n,m}(\rho)$ and a circular harmonic. Therefore (2.18) can be rewritten as

$$F_I(n,m) = \int_0^{\infty} \rho R^*_{n,m}(\rho) \times \left[\frac{1}{\sqrt{2\pi}}\int_0^{2\pi} I(\rho,\theta)e^{-jm\theta}d\theta\right] d\rho \tag{2.20}$$

The integral in square brackets, let us call it $\widehat{I}(\rho)$, is the Fourier series of $I(\rho,\theta)$ along the angle coordinate. Therefore, a rotation of θ_0 radians in I contributes just a phase term $e^{jm\theta_0}$ in $\widehat{I}$, which disappears if one takes the magnitude of the coefficients, thereby obtaining rotation invariance.

The various CHTs differ in the radial profile. We consider three choices, the Zernike Moments (ZM) [97], the Polar Cosine Transform (PCT) [105], and the Fourier-Mellin Transform (FMT) [93]. Zernike radial functions are defined as

$$R_{n,m}(\rho) = \sum_{h=0}^{(n-|m|)/2} C_{n,m,h}\rho^{n-2h} \tag{2.21}$$

for $\rho \in [0,1]$, with $C_{n,m,h}$ suitable coefficients that ensure orthonormality of the basis functions. Fig.2.6(a) shows some of these functions, chosen among those with lowest order. In the PCT, the radial functions are just cosines with argument ρ^2,

$$R_n(\rho) = C_n \cos(\pi n \rho^2) \tag{2.22}$$

limited again to $\rho \in [0,1]$, with normalizing coefficients C_n, some of which are shown in Fig.2.6(b). In the FMT, instead, they are defined as

$$R_\nu(\rho) = \frac{1}{\rho^2}e^{j\nu \ln(\rho)} \tag{2.23}$$

Notice however that, in this case, the functions are non-zero for all $\rho \geq 0$, they diverge at the origin, and the parameter ν is continuous-valued. With this choice, the integral (2.20) becomes just the Fourier transform of $\widehat{I}(\rho)$

after a coordinate remapping, while the whole FMT can be regarded as the bi-dimensional Fourier transform of I in log-polar coordinates. As a consequence, a scale change in I contributes only a phase in the FMT coefficients, which disappears after taking the absolute value, granting also scale invariance.

Now, we have to translate these theoretical definitions into practical finite-length features which characterize locally the image. These must be computed on the available data, sampled on a discrete grid, preserving the invariance properties. To this aim, we have to select a finite number of (n, m) couples, define a suitable patch size and, for each pixel s, compute the $F_{I(s)}(n, m)$ coefficients, by approximating the integral of (2.18) with a summations over the patch centered on s. Eventually, the feature $f(s)$ will be the collection of the magnitudes of these coefficients.

The patch size must guarantee a good compromise between discrimination and robustness. Patches too small might not catch the local image behavior, while if too large they might loose resolution and lead to false alarms. Likewise, features should not be unnecessarily long, to avoid slowing down all processing steps, but still expressive enough to allow correct matches. We will not indulge in describing the preliminary experiments carried out to set these quantities, selected values are reported in Tab.2.1. Patches of 16×16 or 24×24 pixels are used, with features of length 12 for Zernike, 10 for PCT, and 25 for FMT, corresponding always to the lower order[2] basis functions.

Let us focus, instead, on the approximation of the integral (2.18). A straightforward solution is to resample the basis functions $K_{n,m}(\rho, \theta)$ on the grid points (x, y) of the analysis patch, W, where the image is defined (see Fig.2.7(a)), computing therefore

$$F'_I(n, m) = \sum_{(x,y)\in W} I(x, y)\, K^*_{n,m}(\rho(x, y), \theta(x, y)) \tag{2.24}$$

with $\rho(x, y) = \sqrt{x^2 + y^2}$ and $\theta(x, y) = \pm \arctan(y/x)$. However, an equally viable solution is to resample the image on polar (or logpolar) coordinates (see Fig.2.7(b)-(c)), and compute

$$F''_I(n, m) = \sum_{(\rho,\theta)\in W} I(x(\rho, \theta), y(\rho, \theta))\, K^*_{n,m}(\rho, \theta)\rho^i \tag{2.25}$$

with $i = 1$ for the polar grid and $i = 2$ for the logpolar one. This seemingly minor difference has non-negligible consequences on performance, in particular on rotation invariance [102]. In fact, polar sampling guarantees perfect

[2]For FMT, $\nu = 2n\pi/\log(\rho_{\max}/\rho_{min})$, for $n = 0, \pm 1, \pm 2$.

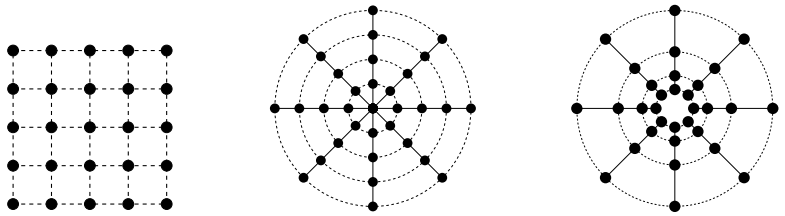


Figure 2.7: Examples of rectangular (a), polar (b) and log-polar (c) sampling grids.

invariance for rotation angles multiple of the sampling step $\Delta\theta$, and a good approximation of it in all other cases, provided $\Delta\theta$ is not too large. On the contrary, with rotation angles close to $\pi/4 \pm k\pi/2$, features computed on the cartesian grid can change significantly, undermining the invariance property, as also shown in [72] where an accurate analysis of errors induced by sampling is carried out.

In addition, the two solutions have the same computational efficiency, since, given ρ and θ, the interpolated values $I(\rho, \theta)$ in (2.25) are computed from available data points with fixed weights, falling back again to a filtering of the form (2.24), only with different weights. We will therefore resort to the polar sampling for both Zernike and PCT features, but keep also the cartesian sampling as reference. For FMT, instead, we will obviously use a log-polar sampling, aiming at scale invariance. However, we are forced to exclude points too close to the origin [115], that is to the central pixel s, where the radial functions diverge.

We note explicitly that CHT-based features have been already used for forgery detection. Zernike moments, for example, have been adopted in [92, 29, 91], with cartesian sampling, providing interesting results. Likewise, the PCT has been investigated in [71], again with cartesian sampling. As for FMT-based features they have been also already considered for forgery detection [11], but with unimpressive results, as reported in [29]. However, the implementation proposed in [11], inspired by [73], includes further processing steps that disrupt the invariance properties, so useful for robust copy-move detection. Similarly, in [100], the features are formed by taking some cross-spectra, rather than the magnitude of the coefficients themselves.

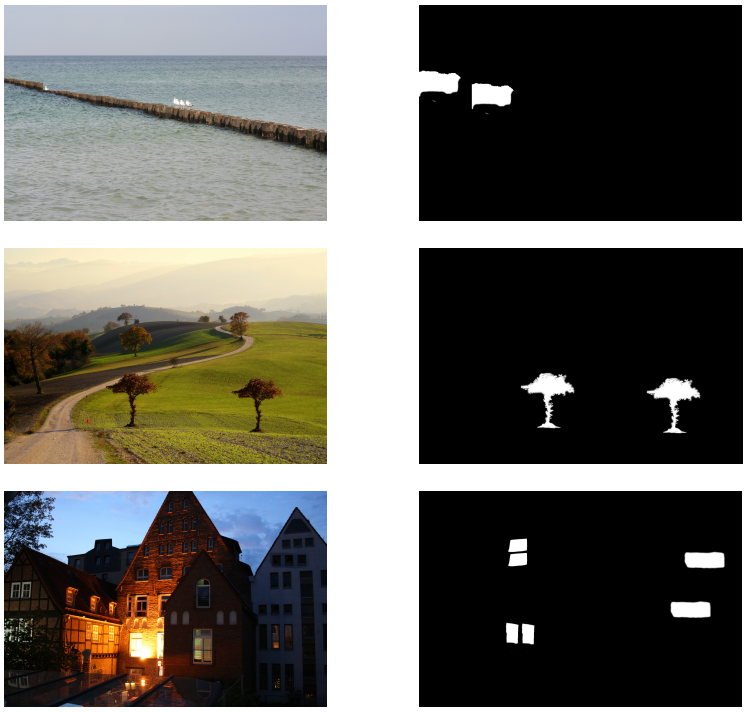


Figure 2.8: Three forged images with different levels of activity from the FAU database. From top: smooth, rough, structured.

2.3 Experimental evaluations

In this section, we present the results of a number of experiments carried out in order to fine tune the proposed technique and assess its performance w.r.t. the state of the art. In order to guarantee reproducibility of results, our code is available online[3], and experiments are carried out on two databases also available online. The database used in [29], which we will call FAU[4] from now on, comprises 48 images with realistic copy-move forgeries, some examples of which are shown in Fig.2.8, classified as smooth, rough or structured. These images are quite large, with typical size 3000×2400 pixels, with tampered areas covering about 6% of each image, on average. We prepared a further database[5] composed by 80 images, again with realistic copy-move forgeries,

[3] http://www.grip.unina.it
[4] http://www5.cs.fau.de/
[5] http://www.grip.unina.it

some of which are shown in Fig.2.9. All these images have size 768×1024 pixels, while the forgeries have arbitrary shapes, aimed at obtaining visually satisfactory results, with size going from about 4000 pixel (less than 1% of the image) to about 50000 pixels. In adding this new database, called GRIP from now on, we wanted to enrich the experimental set available to the community, and include also forgeries of relatively small size. However, we were also motivated by the practical need to run in a reasonable time the large number of experiments needed to fine-tune and validate the proposed technique in various situations of interest.

Results are provided both at pixel level and image level. To assess synthetically the image-level performance we use the F-measure, defined as

$$\mathrm{F} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{2.26}$$

where TP (true positive), FN (false negative), and FP (false positive) count, respectively, the number of detected forged images, undetected forged images, and wrongly detected genuine images. Similar definitions are used at pixel-level for each image to obtain, after averaging on all images, the pixel-level F-measure. At image level we measure, therefore, only the ability to correctly recognize an image as forged or genuine, while the pixel-level measure accounts also for localization accuracy. At pixel level we exclude from computation the pixels at the boundary between forgery and background, where the transparency is set to an intermediate value between 0 and 1 to avoid artifacts. Processing time is another key performance parameter, since reliable copy-move detectors are known to be rather slow, a non-negligible problem with images that become larger and larger as technology goes on. We will therefore report also the average CPU time per image, measured on a computer with a 2GHz Intel Xeon processor, operating in single-thread modality.

Next subsection is devoted to analyze the proposed technique, while the subsequent one compares performance with the state of the art.

2.3.1 Fine tuning of the proposed method

In the proposed technique there are a number of design choices and numerical parameters to be defined. After some preliminary experiments, we set all parameters to the reasonable and non-critical values reported in Tab.2.1. The number of PatchMatch iterations, N_{it}, however, deserves a deeper analysis, since it can impact significantly on the overall performance. We pointed out

Figure 2.9: Three forged images from the GRIP database with different levels of activity. From top: smooth, mixed, textured.

already that a good offset field can be obtained after a small number of iterations, but how small is "small", exactly, and what are the effects of this choice on performance? To gain insight into this point, we ran a series of experiments on our 80-image database, considering various situations of interest: noise addition, JPEG compression, resizing and rotation. In these experiments, we used the Zernike features with polar sampling, with the default parameters of Tab.2.1. Results on accuracy are reported in Fig.2.10, and show that the performance is already very good with as little as 4 iterations, with the only exception of rotated forgeries, where increasing N_{it} to 8 or even 16 guarantees some improvements, up to 0.1, for large angles. In Tab.2.2 we report instead the CPU time. This is constant for feature extraction and nearly so for post-processing, while it grows almost linearly with N_{it} for the matching

param.	value	phase	meaning
$\|W\|_C$	16×16	F	(x, y) samples in cartesian grid
$\|W\|_P$	26×32	F	(ρ, θ) s.s in polar grid ($\rho \leq 8$)
$\|W\|_{LP}$	26×32	F	(ρ, θ) s.s in logpolar grid ($\rho \leq 12$)
$\|f\|_{\mathrm{Z}}$	12	F	length of Zernike features
$\|f\|_{\mathrm{PCT}}$	10	F	length of PCT features
$\|f\|_{\mathrm{FMT}}$	25	F	length of FMT features
N_{it}	4-16	M	# PatchMatch iterations
T_{D1}	8	M	minimum length of offsets
T_{D2}	50	PP	minimum distance between clones
T_{ϵ}^2	300	PP	threshold on DLF error
T_S	1200	PP	minimum size of clones
ρ_M	4	PP	radius of median filter
ρ_N	6	PP	radius of DLF patch
ρ_D	10	PP	radius for morphological dilation

Table 2.1: Relevant parameters for the various phases of the PM-based technique and proposed setting.

phase, which weights heavily on the overall efficiency of the algorithm. With $N_{it} = 4$, the proposed algorithm takes about 11 seconds/image, on the average, a time that almost doubles with $N_{it} = 16$. Larger values make no sense as they have no effect on accuracy. Therefore, the user can be interested in several profiles, from FAST ($N_{it} = 4$) with an overall processing time competitive with that of keypoint-based techniques, but a much better accuracy, to ACCURATE ($N_{it} = 16$), which guarantees the best performance at the cost of a longer, but still reasonable, CPU time.

The other major choice available to the user concerns the type of feature to use. Again, for each feature we set in advance the main parameters as reported in Tab.2.1. The pixel-level curves shown in Fig.2.11, obtained with $N_{it} = 8$, are quite close to one another. With all features, the proposed technique behaves very well on rigid copy-moves, with an F-measure going from 0.90 for FMT to 0.94 for Zernike-polar. Moreover, the performance degrades in a similar manner for all features with increasing noise, compression ratio, rescaling factor, and rotation angle. Some minor differences can be observed for FMT, slightly better than the others for moderate rescaling factors, and for

N_{it}	Feat. Ext.	Matching	Post-Proc.	Total
2	2.36 (0.48)	3.19 (0.32)	1.93 (0.18)	7.48 (0.66)
4		6.54 (0.96)	2.12 (0.30)	11.02 (1.49)
8		11.96 (1.68)	2.10 (0.25)	16.42 (2.00)
16		16.81 (1.51)	1.77 (0.31)	20.92 (1.85)

Table 2.2: CPU-time performance (seconds/image) vs N_{it} using Zernike-polar features; in parentheses the standard deviations.

Feature	Feat. Ext.	Matching	Post-Proc.	Total
ZM-cart	2.55 (0.52)	12.00 (1.70)	2.08 (0.25)	16.63 (2.02)
ZM-polar	2.36 (0.48)	11.96 (1.68)	2.10 (0.25)	16.42 (2.00)
PCT-cart	1.79 (0.29)	10.78 (1.48)	2.30 (0.32)	14.86 (1.66)
PCT-polar	1.74 (0.24)	10.79 (1.25)	2.16 (0.29)	14.69 (1.40)
FMT	9.82 (1.47)	15.23 (1.72)	1.80 (0.16)	26.86 (2.83)

Table 2.3: CPU-time performance (seconds/image) vs feature using $N_{it} = 8$; in parentheses the standard deviations.

Zernike-cartesian, slightly worse at large-angle rotations.

With these results, processing time becomes again a key decision element. Tab.2.3 shows the average CPU-time for the selected features with the default parameters of Tab.2.1. It results that only the FMT feature presents a significantly different (higher) processing time, due to the larger patch used to compute the feature, and the longer features themselves.

Before comparing results with the state of the art, we present some experiments to assess the impact of each of the proposed improvements. In particular, keeping fixed the parameters selected before and using the Zernike-polar features when applicable, we consider a baseline reference technique with basic PatchMatch and SATS, two intermediate versions where only the matching phase or the post-processing are improved, switching to our modified PatchMatch or to the proposed post-processing based on dense linear fitting, respectively, and then the proposed technique including both improvements. Results are reported in Fig.2.12 and Tab.2.4. All versions provide a similar accuracy performance in the presence of noise, compression and resizing, except for a small but consistent improvement observed when the DLF-based post-processing is used. In the presence of rotation, instead, for angles be-

Version	Feat. Ext.	Matching	Post-Proc.	Total
baseline	2.06 (0.21)	9.00 (0.84)	82.68 (59.61)	93.74 (59.51)
+modPM	2.36 (0.48)	11.96 (1.68)	91.77 (57.23)	106.09 (56.95)
+DLF	2.06 (0.21)	9.00 (0.84)	2.01 (0.27)	13.07 (0.99)
proposed	2.36 (0.48)	11.96 (1.68)	2.10 (0.25)	16.42 (2.00)

Table 2.4: CPU-time performance (seconds/image) vs version; in parentheses the standard deviations.

yond 15 degrees the performance drops sharply for the versions using basic PatchMatch, while it remains almost constant for those using modified PatchMatch. As for computational efficiency, the basic and modified versions of PatchMatch are almost equivalent, while SATS is much slower than DLF, and responsible for most of the overall running time, when used. SATS complexity descends from its iterative nature [28]: given a few reliable close points, it estimates an affine transform explaining their offsets, and gradually enriches the set including only points obeying the same transform. Therefore, outliers are automatically rejected, improving robustness at the cost of higher CPU time. With PatchMatch, however, the offset field is already quite regular, allowing for the use of the much faster DLF without harm.

Technique	F-image	F-pixel	CPU
Christlein2012	93.20	93.52	4377.60
Bravo2011	96.97	90.52	2149.91
Amerini2013	74.07	50.11	156.88
Cozzolino2014	94.85	89.77	2366.57
PM-ZM-cart	93.07	93.11	287.50
PM-ZM-polar	94.95	93.72	244.67
PM-PCT-cart	94.00	93.17	281.37
PM-PCT-polar	95.92	93.62	293.84
PM-FMT	92.00	89.46	424.26

Table 2.5: Image-level and pixel-level F-measure and total CPU-time for rigid copy-moves on the FAU database.

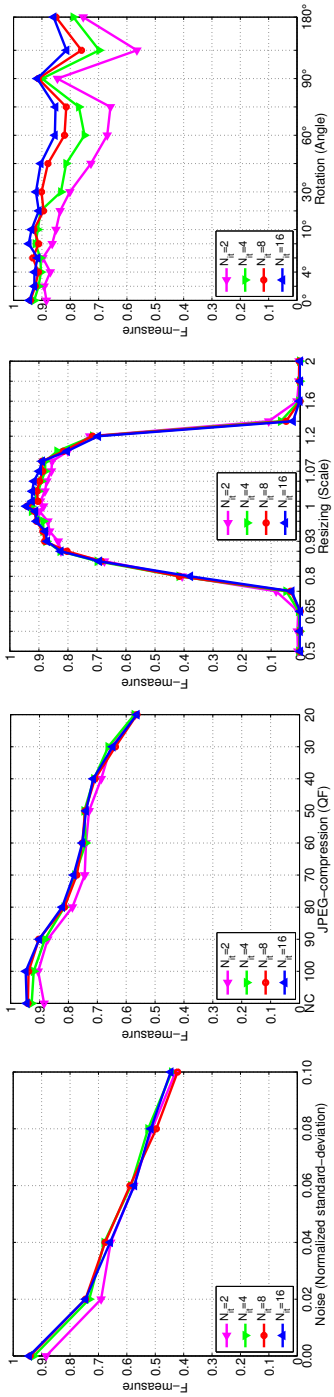


Figure 2.10: Pixel-level F-measure curves for the proposed PM-based technique (Zernike-polar feature) for different values of N_{it}.

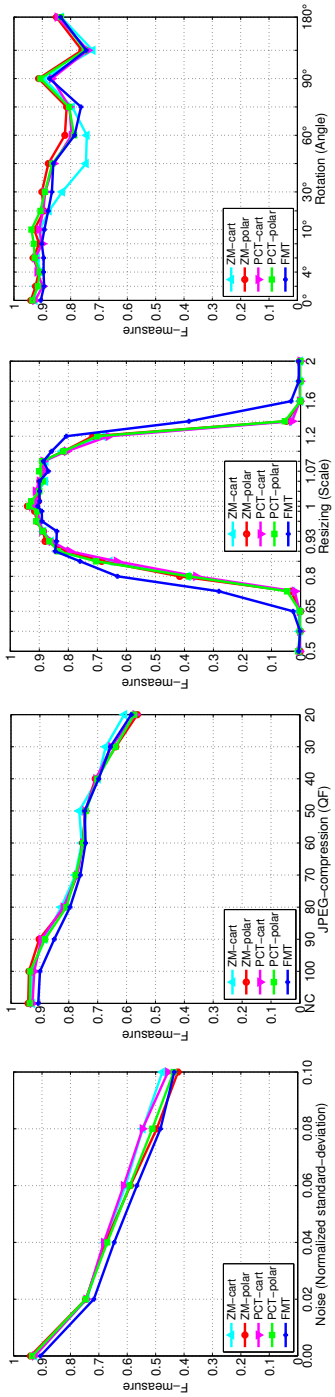


Figure 2.11: Pixel-level F-measure curves for the proposed PM-based technique ($N_{it} = 8$) with different features.

2.3.2 Comparison with the state of the art

To position the proposed approach w.r.t. the current state of the art, results are compared with those of several promising techniques recently proposed in the literature, most of them designed to deal also with rotation and rescaling. In particular we consider Bravo2011 [16], Christlein2012 [29][6], Amerini2013 [6], Ryu2013 [91], and Cozzolino2014 [35], the latter being a previous version of the technique proposed here, with Zernike-cartesian feature, SATS postprocessing, and other minor differences. A comparison with the other techniques reviewed in [29] can be established using (with caution) the transitive property. Unfortunately, Ryu2013 turned out to be exceedingly slow on smooth images (about 15 hours on the 768×1024 image on the top of Fig.2.9), so it does not appear in the following results, and we will design a separate experiment to compare with it. For the proposed PatchMatch-based technique (shortnamed PM from now on) we set $N_{it} = 8$ once and for all, but test all proposed features.

In Tab.2.5 and Tab.2.6 we report, for the FAU and the GRIP databases, respectively, the image-level and pixel-level F-measure, together with the total CPU time, observed for rigid translation. In the top part of the tables we group reference techniques, and in the bottom part the various versions of the proposed one. Taking into account the obvious differences, results are very well aligned on the two databases. In particular, the proposed technique performs best with polar features, both Zernike and PCT, being competitive with the other dense-field techniques at image level, and generally better than them at pixel level. As expected, the only keypoint-based technique, Amerini2013, provides a much worse detection performance, but also the smallest processing time. Among dense-field techniques, PM is by far the most efficient, with all features. With Zernike-polar features, in particular, it is at least 3 times faster than the dense-field references on the GRIP database, and at least 9 times faster than them on the FAU database. The different speeding-up factors suggest that, while the complexity of the PatchMatch-based technique scales almost linearly with the image size, this might not hold for the other dense-field techniques.

We now test robustness against noise addition, compression, rotation and resizing, showing the image-level and pixel-level F-measure curves in Figg.2.13 and 2.14, respectively. For the sake of clarity, only the Zernike-polar feature is considered for the proposed technique. At image level, as already reported in the tables, the dense-field Bravo2011 and Christlein2012

[6]We refer to the technique with Zernike features, kd-tree matching, and SATS post-processing.

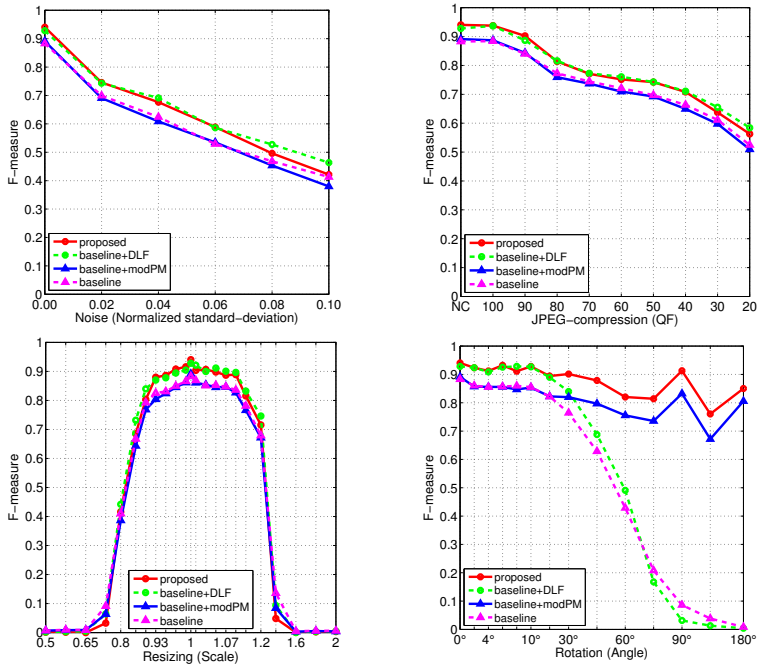


Figure 2.12: Pixel-level F-measure curves for the proposed technique, the baseline reference, and intermediate versions.

exhibit the best performance in the ideal case, a performance which degrades rapidly, however, with increasing levels of noise and JPEG compression. At pixel-level, instead, the proposed PatchMatch-based detector outperforms consistently all the references, with a performance gain that becomes very significant in the presence of intense noise and large compression factors, as well as for moderate scale changes and critical rotation angles around 45^o and 135^o.

The SIFT-based Amerini2013 deserves a separate discussion. In fact, thanks to the use of keypoints, it provides the most stable performance across all conditions, including large-scale resizing, where all considered dense-field techniques fail. Unfortunately, its overall performance is doomed by the inability to discover copy-moves in smooth areas, lacking the keypoints, and localization seems to be quite imprecise. Given these complementary properties, a suitable fusion of keypoint and dense-field approaches may be expected to provide good results.

Let us now describe the results of an *ad hoc* experiment designed to in-

Technique	F-image	F-pixel	CPU
Christlein2012	98.16	87.44	53.91
Bravo2011	95.81	84.44	102.78
Amerini2013	67.72	44.41	3.71
Cozzolino2014	94.67	88.67	54.74
PM-ZM-cart	92.94	92.66	16.63
PM-ZM-polar	94.05	94.06	16.42
PM-PCT-cart	94.61	92.55	14.86
PM-PCT-polar	94.05	93.51	14.69
PM-FMT	91.33	90.56	26.86

Table 2.6: Image-level and pixel-level F-measure and total CPU-time for rigid copy-moves on the GRIP database.

clude the promising Ryu2013 technique, based on LSH and RANSAC, in the comparative analysis. To this end, we built a database comprising 40 textured images, a case in which Ryu2013 presents an acceptable CPU-time, with size 512×512-pixel, and random square forgeries. Fig.2.15 shows the image-level F-measure curves. On this set of images, the proposed technique and Christlein2012 provide almost perfect detection in all cases, except for large rescaling and intense noise. Also the performance of Ryu2013 is quite stable, but starting from a worse result on rigid copy-moves. This is due to the large number of false alarms observed in genuine images. In fact, at pixel level, when only forged images are considered, Ryu2013 performs as well as the proposed technique as clear from Fig.2.16. However, it spends on average 159.05 seconds to process these images, much more than the 21.27 and 4.70 seconds necessary, respectively, for Christlein2012 and for the proposed technique.

We conclude this Section by showing some examples of challenging copy-moves from the GRIP database together with the color-coded detection mask output by Christlein2012 and by the PatchMatch-based detector. Christlein2012 detects all forgeries, but the masks are rather inaccurate. Moreover, in the second image there are a few false alarms due to a flat background. On the contrary, the proposed technique detects all forgeries with remarkably accurate masks and without false alarms.

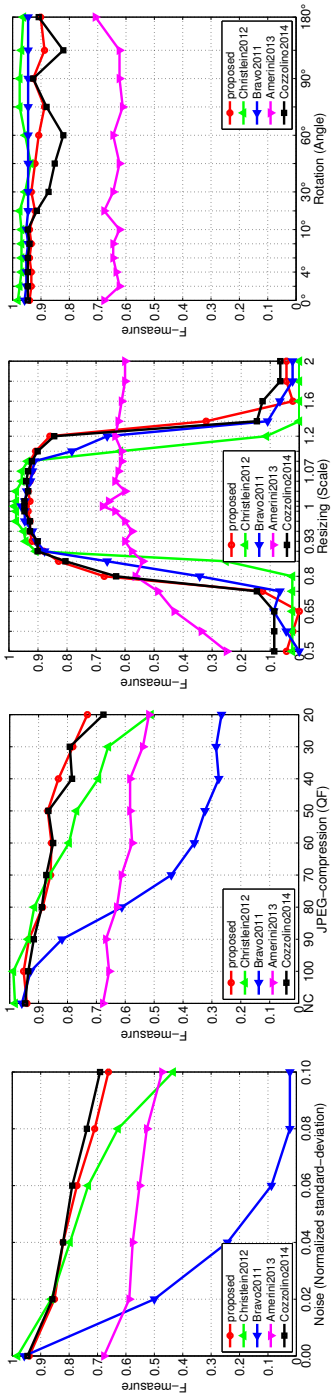


Figure 2.13: Image-level F-measure curves for the proposed (PM-ZM-polar) and reference techniques.

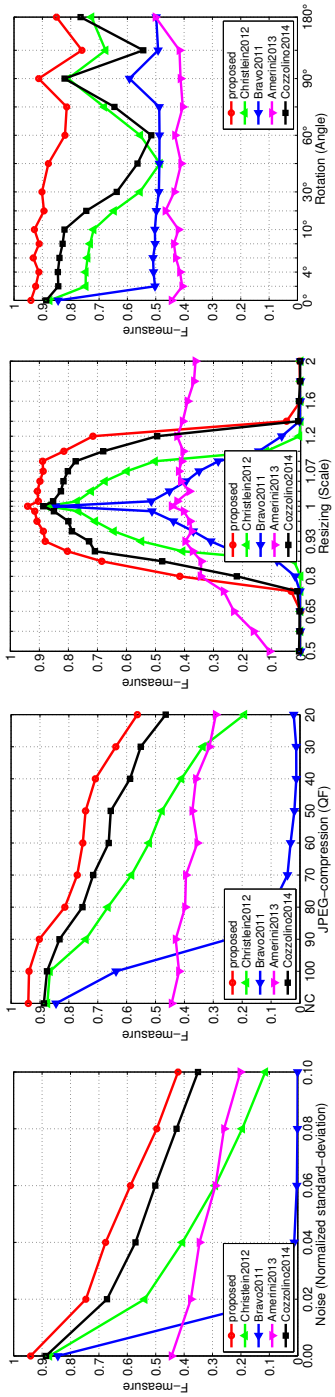


Figure 2.14: Pixel-level F-measure curves for the proposed (PM-ZM-polar) and reference techniques.

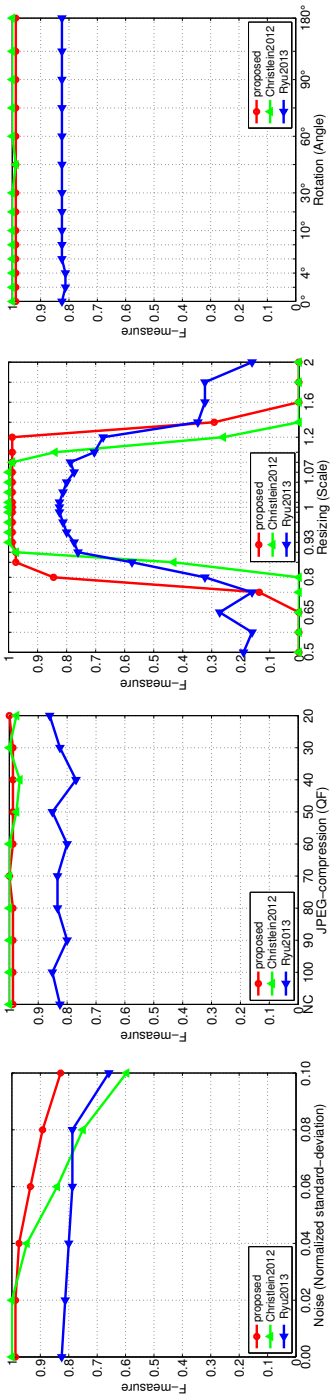


Figure 2.15: Image-level F-measure curves for some selected techniques on textured images.

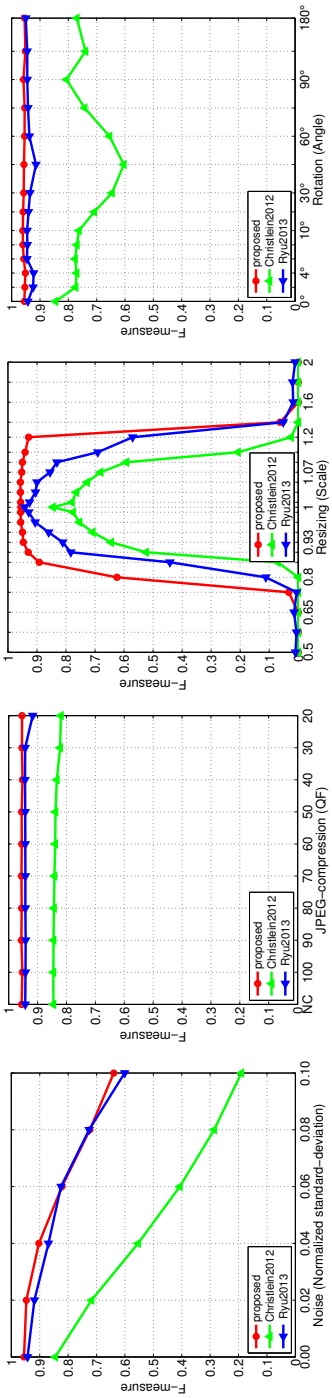


Figure 2.16: Pixel-level F-measure curves for some selected techniques on textured images.

2.4 Conclusions

Copy-move forgeries are extremely common, and can be carried out easily and accurately. Detecting them, on the contrary, can be quite challenging, especially if they are of the occlusive type, with pieces of background copied elsewhere to hide some subjects of interest. In these cases, keypoint-based techniques are mostly ineffective, as they totally neglect low-entropy background areas. Dense-field techniques, on the other hand, tend to be very slow because of the feature matching phase, a problem which becomes worse and worse as the average image size increases.

The technique proposed here is a first step towards fast and accurate copy-move detection. We use the PatchMatch algorithm to compute efficiently a high-quality approximate nearest neighbor field for the whole image. Given this major achievement, we then reduce the overall complexity by implementing also a fast post-processing procedure based on dense linear fitting. Moreover, by resorting to state-of-the-art invariant features, and a suitably modified version of PatchMatch, we achieve also a good robustness to various type of geometrical distortions.

Experimental results are quite satisfactory, as they show the proposed technique to provide state-of-the-art detection performance, with significant improvements in terms of localization accuracy and speed. Nonetheless, there is much room for further improvements. PatchMatch is certainly effective, but new fast matching algorithms are proposed by the day, and further progresses can be easily foreseen. Likewise, our post-processing, though effective, can be certainly further refined. Even so, with much larger images, which will probably become customary in the near future, all these tools may soon become ineffective, and multiresolution analysis is probably a more solid path for future research.

Turning to accuracy, a major goal is to achieve higher robustness to resizing, and include also other forms of geometric distortion. Under this point of view, the discrete-domain FMT features fail to guarantee the invariance properties promised by their continuous-domain counterparts. Better implementations of FMT, or completely new and more robust features, can be probably proposed. Moreover an extinction of the technique to analyse forged video would be interesting.

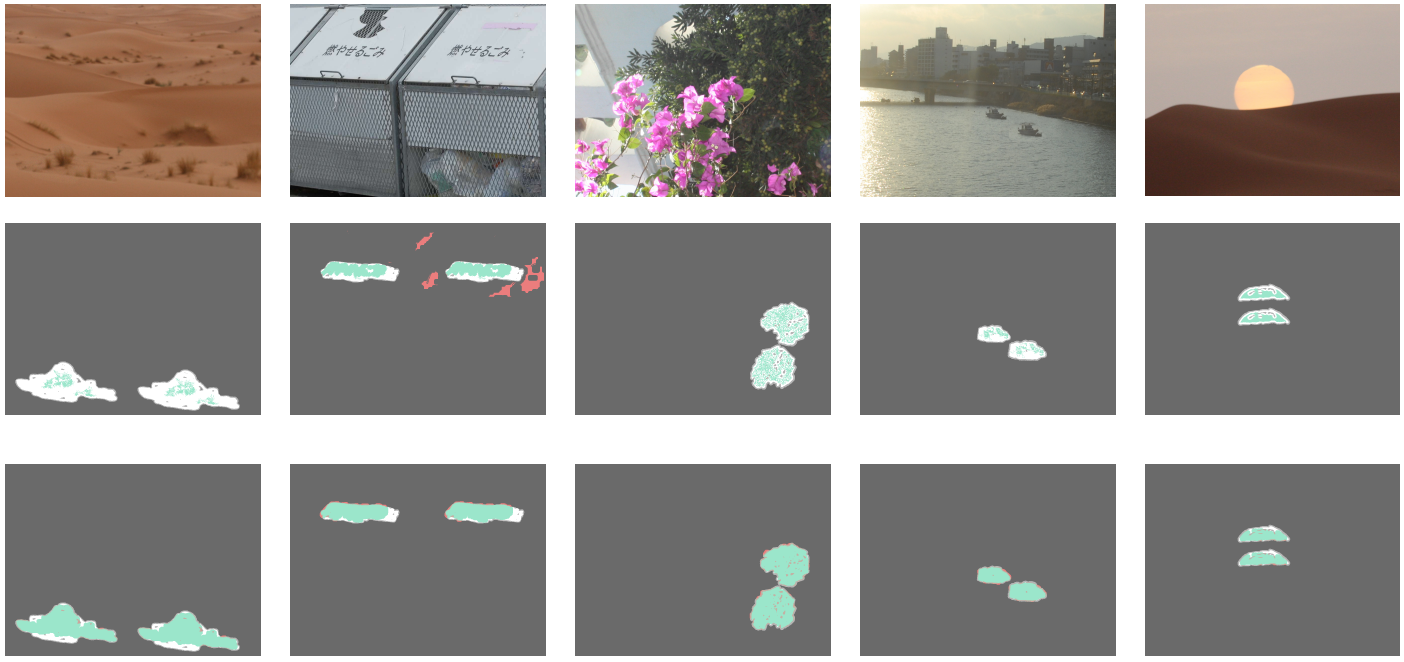


Figure 2.17: Forgery detection masks for some images of the GRIP database. From top to bottom: forged images, masks output by Christlein2012, masks output by the proposed method. Green indicates correct detection, false alarms are in red. From left to right: noise with normalized std 0.02, JPEG compression with Q=60, rotation with $\theta = 45^o$, rescaling with $\alpha = 1.145$, occlusive rigid copy-move.

Chapter 3

Feature-based approach for forgery localization

In this chapter a new camera-based technique based on a *dense local descriptor* is proposed for tampering localization [99]. Local image descriptors have by now reached a prominent status in image processing. They have been used with success for such diverse and challenging tasks as image mining and retrieval [113], texture classification [85, 98], face recognition [4, 18], fingerprint liveness detection [52, 51], steganalysis [44], image quality assessment [81]. In the proposed method, a local descriptor is extracted for each block of the image under analysis and compared with a model estimated in advance. Based on the fitting with the model, a score is computed and aggregated over neighboring blocks, to obtain a map highligting likely forged regions. Model estimation is a crucial step, which requires the availability of the camera or of a sufficient number of pristine images taken by it.

3.1 Introduction

In forgery detection [56], the key idea is that suitable features can capture the deviations from the normal behavior induced by typical image forgeries, such as splicings. It is worth underlining that these deviations are often not perceivable by a human being, since modern image editing tools, if used with proper skill, allow one to manipulate images leaving little or no obvious artifacts, smoothing the boundary between host image and forgery to avoid abrupt transitions. Major efforts have been devoted to find good statistical models for natural images in order to select the features that guarantee the highest discrim-

inative power. Often, in order to capture more meaningful statistics, transform-domain features have been used, as in [94] where the image undergoes block-wise discrete cosine transform (DCT) with various block sizes and first-order (histogram based) and higher-order (transition probabilities) features are collected and merged.

Recently, following an approach used in steganalysis [44], we proposed [31] a powerful descriptor-based forgery detection technique. A high-pass filtering is first carried out to compute a residual image where the useless high-level information is removed and anomalies can be better detected. Then, synthetic features are computed by means of a histogram of occurrence. Given the good results obtained in detection, we designed also a sliding-window version of the same algorithm [32] devoted to forgery localization. Specifically, this latter algorithm was designed to detect traces left by splicings at the boundary with the host image. However, by deeper investigation, we realized that the proposed descriptor was discovering much more than the anomalies related to unnatural boundaries. In fact, it was revealing more general deviations from the typical appearance of a natural image.

We considered then the following non-exclusive hypotheses:

1. the algorithm was detecting the different camera (device, model, or brand) that generated the splicing;

2. the algorithm was detecting some forms of image processing.

Indeed, it is well-known [17] that various types of artifacts exist, specific of a manufacturer or a model or even an individual camera which, in suitable hypothesis, enable one to identify the source of a given image. Likewise, when an image is tampered with, the different processing history of its regions can be traced back. Much of the current literature based on features aims to explore some specific types of processing the forgery could have been subject to. In [107], following [64], a feature-based procedure is outlined in order to tell apart regions subject to median filtering from region treated by other forms of processing. An analogous approach is used in [111], where a noncausal Markov model is considered in order to capture the underlying statistical characteristics of the signal. Feature-based classification and localization is also performed in [22], where blurring is detected by using features already considered for the evaluation of natural image statistics in the context of image quality assessment [81]. The key idea is that these statistics change when blurring takes place.

In all these techniques a two-class (pristine/forged) training procedure is necessary and each method focuses on a particular type of manipulation. The method proposed in this chapter is itself feature-based, but is not tailored to a specific type of tampering and requires training only on pristine images. In particular, the proposed method requires the availability of the source camera or else of a good number of pristine images taken by it, which are the typical hypotheses of camera-based methods, like those based on sensor noise [20]. Once local statistics are learnt from the training images, the test image is analyzed in sliding-window modality to discover deviations from the model, and local distance measures are aggregated to build a decision map. Unlike techniques based on sensor noise, the proposed algorithm is not influenced by the scene content, and is computationally efficient.

In next Section the proposed algorithm is described. Sections III and IV are devoted to the experimental validation, conducted first in more controlled conditions, to establish some basic properties of the approach, and then for realistic forgery localization tasks. Finally, the last section draws conclusions.

3.2 Proposed method

In the following we describe how features are extracted, and how they are used for detection and localization.

3.2.1 Feature extraction

We follow the three-step model already used in [44, 31] comprising

1. computation of residuals through high-pass filtering;
2. quantization of the residuals;
3. computation of a histogram of co-occurrences.

The final histogram is the feature vector associated with the whole image, which can be used for classification. To compute the residual image we use a linear high-pass filter of the third order, which assured us a good performance in the context of forgery detection [31], defined as

$$r_{ij} = x_{i,j-1} - 3\,x_{i,j} + 3\,x_{i,j+1} - x_{i,j+2}$$

where x and r are origin and residual images, and i, j indicate spatial coordinates.

More then in the residual themselves, we are interested in their co-occurrences, which provide information on higher-order phenomena and are based on larger support areas. Of course, residuals must be first quantized and, in order to obtain a manageable number of bins in the histogram, a very small number of quantization values must be considered. As suggested in [44] we perform quantization and truncation as:

$$\widehat{r}_{ij} = \mathrm{trunc}_T(\mathrm{round}(r_{ij}/q))$$

with q the quantization step and T the truncation value. To limit the matrix size we use $T = 2$ and $q = 1$. At this point we compute co-occurrence on four pixels in a row, that is

$$C(k_0, k_1, k_2, k_3) = \sum_{i,j} I(q_{i,j} = k_0, q_{i+1,j} = k_1, q_{i+2,j} = k_2, q_{i+3,j} = k_3)$$

The homologous column-wise co-occurrences are pooled with the above based on symmetry considerations, obtaining eventually a 625-bin histogram, which is reduced to little more than 300 by further symmetry arguments.

As a final step, to reduce the weight of outliers in the subsequent training and classification phases, we pass the resulting features through a square-root non-linearity. Starting from normalized histograms (unitary sum) the final features happen to have unitary L2 norm.

3.2.2 Detection/identification

We consider first the simpler detection problem, which requires to classify a whole image or image region as genuine (hypothesis H0) or tampered (hypothesis H1). We assume to know the camera which took the photos and have a large enough collection of images taken with the same camera or even the freedom to take new ones. On the contrary, we know nothing on the camera used to produce the possible forgery. Rather than training a two-class classifier using blocks drawn from the most heterogeneous sources for hypothesis H1, we consider a model-based approach. Following the methodology used in [82], we fit the available H0 samples through a multidimensional gaussian and carry out a threshold test. To this end, we estimate the mean vector and covariance

Figure 3.1: Graphic scheme of aggreation procedure.

matrix of the features $\mathbf{h}_n$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{h}_n$$
$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{h}_n - \boldsymbol{\mu})(\mathbf{h}_n - \boldsymbol{\mu})^T$$

Then for each new feature under test, say $\mathbf{h}'$ we compute the log-likelihood w.r.t. the Gaussian model (neglecting constants)

$$L(\mathbf{h}') = (\mathbf{h}' - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{h}' - \boldsymbol{\mu})$$

and compare it with a threshold. Setting the threshold might be a challenging problem, but we do not analyze it here, and will compute performance as a function of this parameter.

3.2.3 Localization

In localization we assume to know already that a region of the image has been tampered with and want to delineate as accurately as possible its position and shape. A simple solution based on the detection procedure described above consists in using it in a sliding-window modality, with the window size $W \times W$ trading off reliability for resolution. We consider partially overlapping blocks, taken with step $1 \leq S < W$, and aggregate all decisions a map, summing -1 or +1 in the map for all pixels of the block depending on the decision, pristine or

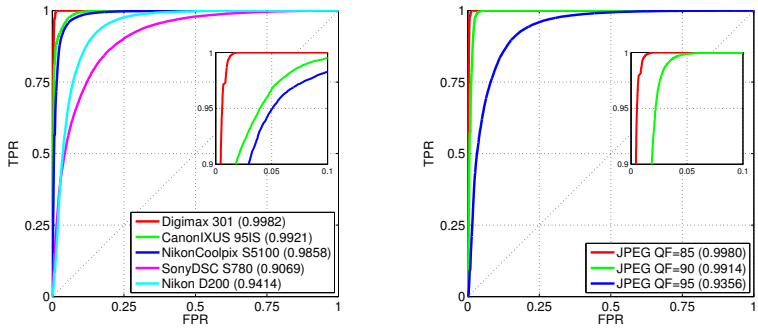


Figure 3.2: Camera-based (left) and processing-based (right) detection performance using a Canon EOS 450D as target camera.

tampered, respectively. To improve the performance, rather than just the sign, we associate a real-valued strength to the block under test, depending on the reliability of the decision, so as to give more importance to clear-cut situations. By aggregating these strengths, a real valued map is generated, based on which all decisions are eventually made. Fig.3.1 describes pictorially the aggregation procedure. In our case, it is straightforward to associate the strength with the log-likelihood itself. A threshold is eventually needed to single out the suspect tampered region but, again, we do not address this problem here, resorting to ROC curves to analyze performance.

3.3 Preliminary tests on tampering detection

Although our main focus is on tampering localization, we carry out some preliminary tests for the more controlled detection case. These tests, in particular, will shed some light on the validity of the two conjectures sketched in the Introduction. We analyze them in turn.

3.3.1 Camera-based detection

For our experiments we have 6 cameras available, of 6 different models, produced by four manufacturers: Canon EOS 450D, Canon IXUS 95IS, Nikon D200, Nikon Coolpix S5100, Digimax 301, Sony DSC S780. For each camera we have a relatively large number of images, always more than 100, which are cropped to size 768×1024, for simplicity, aligned with the JPEG grid.

As first experiment we consider one target camera for hypothesis H0, and

several more for hypothesis H1. From each available test image, not included in the training set, we extract 140 blocks of size 128×128 pixels, drawn with step 64 pixels on rows/columns. Each block is then independently classified as pristine or tampered. In Fig.3.2(left) we show the receiver operating curves (ROC) obtained for the Canon EOS 450D by varying the decision threshold. Results are always very good, and almost perfect in several cases. Although good results can be obtained with other approaches as well, it is worth underlining that our method is very general, does not depend on specific attributes of digital photos (e.g., CFA, quantization tables, etc.) nor is tailored to specific brands or models.

This experiment makes clear that the descriptor is indeed capturing some subtle camera-related feature and hints (a field proof is given in next section) that a splicing coming from a different camera can be very likely detected and localized. Let us therefore turn to the second conjecture: can we detect tampering based on processing history?

3.3.2 Processing-based detection

To test the second conjecture, we now consider the same camera for both host images and forgeries, but assume that the forgery has been subject to some type of processing before splicing, such as JPEG compression, resizing, etc., which is typically the case, both for the different history of the images and because the inserted material is often manipulated to have a natural appearance in the new context. In this case, quite a large number of combinations could be considered, but we focus only on JPEG compression, postponing to next Section a more detailed analysis. Results shown in Fig.3.2(right) are also in this case very good. When the test blocks are compressed with QF 85 or even 90, the ROC is almost perfect, characterized by an area under curve (AUC), shown in parentheses in the legend, exceeding 0.99. Performance becomes appreciably worse for QF 95, but remains still good (with AUC=0.93), considering also that, for the camera under test, blocks that are nominally uncompressed are actually compressed with quality factor 98.

This second experiment fully confirms also our second conjecture, so we can conclude that the proposed approach is able to accurately identify deviations from the model, even subtle deviations, due both to the different source camera and to the different processing history of the forgery.

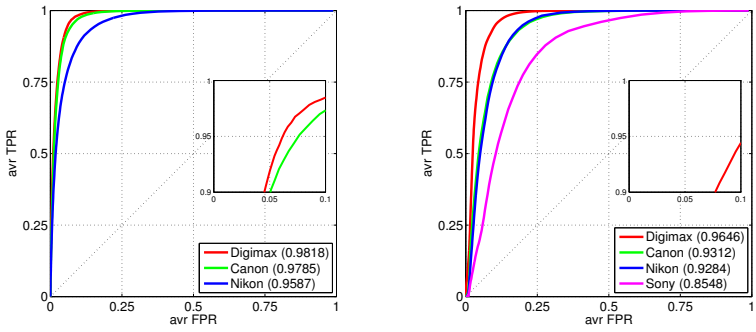


Figure 3.3: Camera-based localization performance (Canon EOS 450D on the left and Nikon D200 on the right).

3.4 Tampering localization experiments

Let us now consider some more realistic experiments with forged images, following the same path of the preceding Section. In each host image, of size 768×1024 pixel, we insert a random square forgery of size 192×192 in random positions in the image.

3.4.1 Camera-based localization

In a first experiment, the host image is taken by the target camera while the forgery comes from another unknown camera. Blocks of size 128×128 are drawn from the image with step 16 pixels on rows/columns. for each one we compute the log-likelihood of its feature w.r.t. the gaussian model fitted to the target camera. These quantities are then reprojected on the image and aggregated. The final map is eventually thresholded, and ROCs are computed as the threshold value λ varies. Fig.3.3 shows the ROCs obtained for two different host cameras (a Canon EOS 450D and a Nikon D200), again quite satisfactory. Note that all performance indicators are computed pixel-wise, therefore the curves depart from the ideal behavior not because forgeries are not localized, which never happens in these experiments, but because of the obvious inaccuracy in detecting the exact shape of the forgery. Some examples are shown in Fig.3.6, last column: the forgeries are correctly localized, but their shape is recovered only approximately. Better decision strategies, as in [23], can certainly improve upon this basic procedure.

3.4.2 Processing-based localization

We now repeat the above experiment, but the forgery is generated by the same camera that took the host image. However, before insertion, the forgery undergoes some kind of processing which changes its statistics. In Fig.3.5 we report some results for two host cameras and four common processing tasks, that is blurring, compression, resizing, rotation. For each case, we show several ROCs obtained by changing the main parameter of interest, e.g., the quality factor in JPEG compression. Results are again very good in all cases. To observe a significant drop in performance we must consider very challenging situations, such as rotation with a very small angle, or JPEG compression with quality factor 95, and even in these cases the AUC remains near or beyond 0.80. It is worth underlining that a human being would hardly detect such high-quality forgeries by visual inspection. Some interesting phenomena, yet to investigate, concern resizing, less detectable for scales below 1 than above it, and rotation, where performance depends more strongly than expected on the angle.

3.4.3 Performance comparison

We conclude this analysis carrying out an experimental comparison with other two well-known camera-based techniques, which exploit the photo response non-uniformity (PRNU) noise, and the color filter array (CFA), respectively. In particular, for PRNU-based localization we use the algorithm recently proposed in [27], using a Bayesian framework and modern optimization techniques, while the CFA-based technique has been proposed in [40]. We built a quite varied training set, using three cameras (Canon IXUS-95IS, Nikon Coolpix-S5100, Digimax-301), and considering all combination of forgeries with and without JPEG compression, resizing, rotation, and blurring. Results are reported in Fig.3.4. The proposed method performs slightly better than the PRNU-based technique, and significantly better than the CFA-based one. Moreover, while the proposed method has negligible complexity, the PRNU-based technique, based on MRF modeling and optimization, has a significant run time.

Finally, we show some examples of realistic forgeries to enable a comparison of the proposed and PRNU-based techniques by visual inspection. Note that the threshold for the proposed method has been set to a fixed value equal to 1.75, while for the PRNU-based technique we considered the setting of the original paper [27]. The first forgery has not been processed, but come from a

camera different from the host, the following two come from the host camera, but have been resized before splicing, the forth and fifth come from a different camera with resizing, the last one from the host camera after blurring.

The proposed technique has a very good detection performance, with no false alarms, while the reference PRNU-based method occasionally (rarely) exhibits some false alarms and misses, as in some of the examples of Figg. 3.6 and 3.7, selected on purpose. For the proposed technique errors concern only the limited ability to follow the shape of the forgery, affecting the pixel-based ROCs.

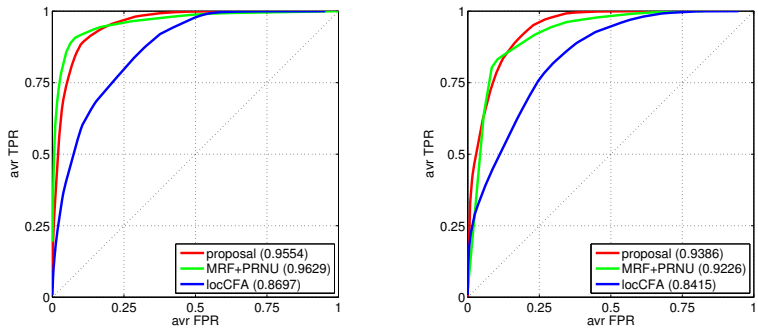


Figure 3.4: Performance comparison (Canon EOS 450D on the left and Nikon D200 on the right).

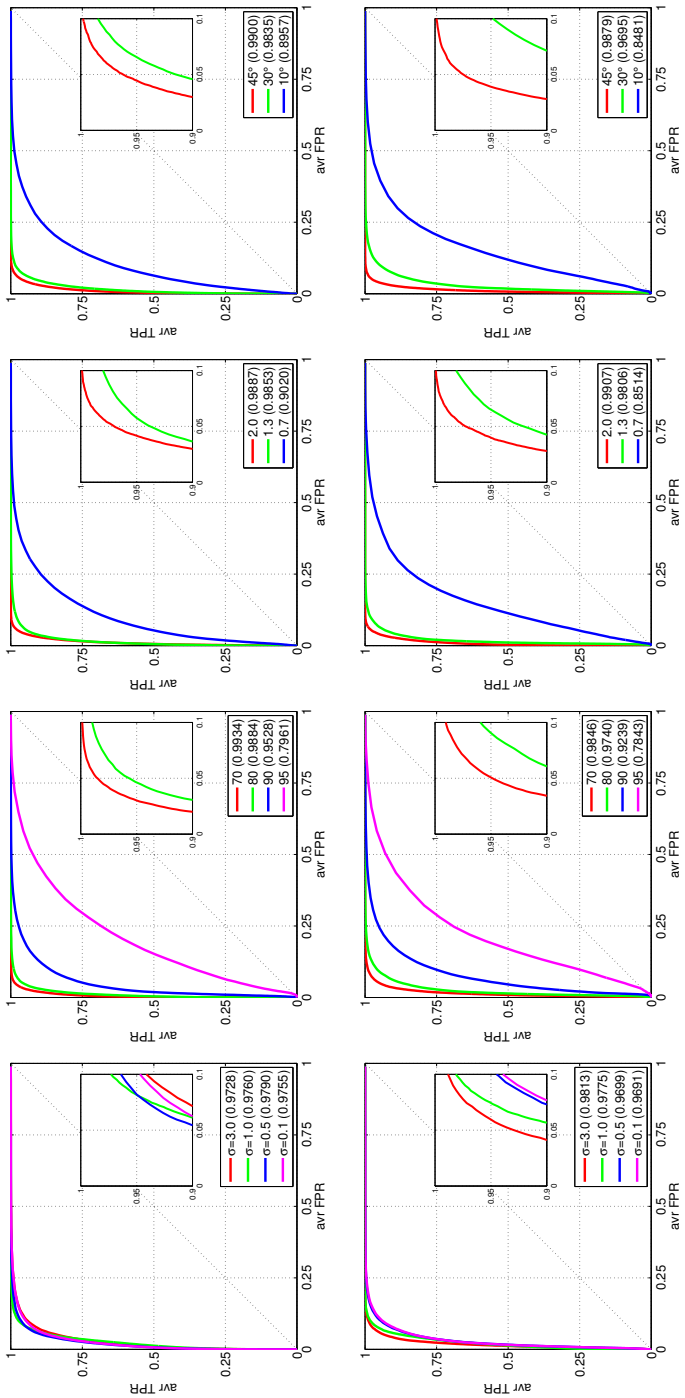


Figure 3.5: Processing-based localization performance (Canon EOS 450D on the first row and Nikon D200 on the second row). From left to right: blurring, JPEG compression, resizing, rotation.

3.5 Conclusions

We propose a new camera-based technique for forgery localization based on a simple modeling of natural image statistics. A large number of blocks are extracted off-line from training images and characterized through features based on a dense local descriptor. A multidimensional Gaussian model is then fit to the training features. In the testing phase, the image is analyzed in sliding-window modality: for each block, the log-likelihood of the associated feature is computed, reprojected in the image domain, and aggregated, so as to form a smooth decision map. Despite its simplicity, it provides a very good performance in a wide range of experimental conditions. Future research will include a more thorough investigation of the many design choices of the technique (e.g. features, model, decision strategy, etc.) and its optimization w.r.t. the main parameters. This analysis could help in understanding which characteristics of the camera are effectively captured by this approach. Further experimental analysis is also necessary to study robustness to subsequent processing and possible countermeasures.

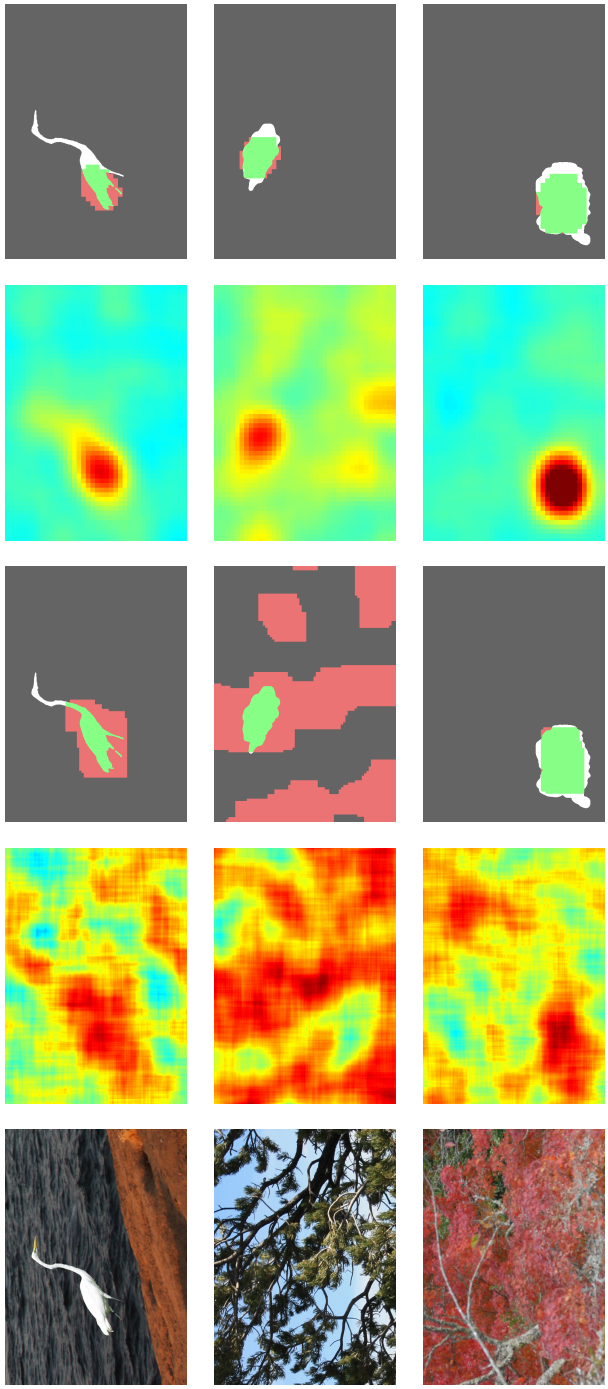


Figure 3.6: Forgery localization results for some selected examples. From left to right: forged image, PRNU correlation index field and color-coded detection mask, proposed aggregation map and color-coded detection mask. Green indicates correct detection, false alarms are in red.

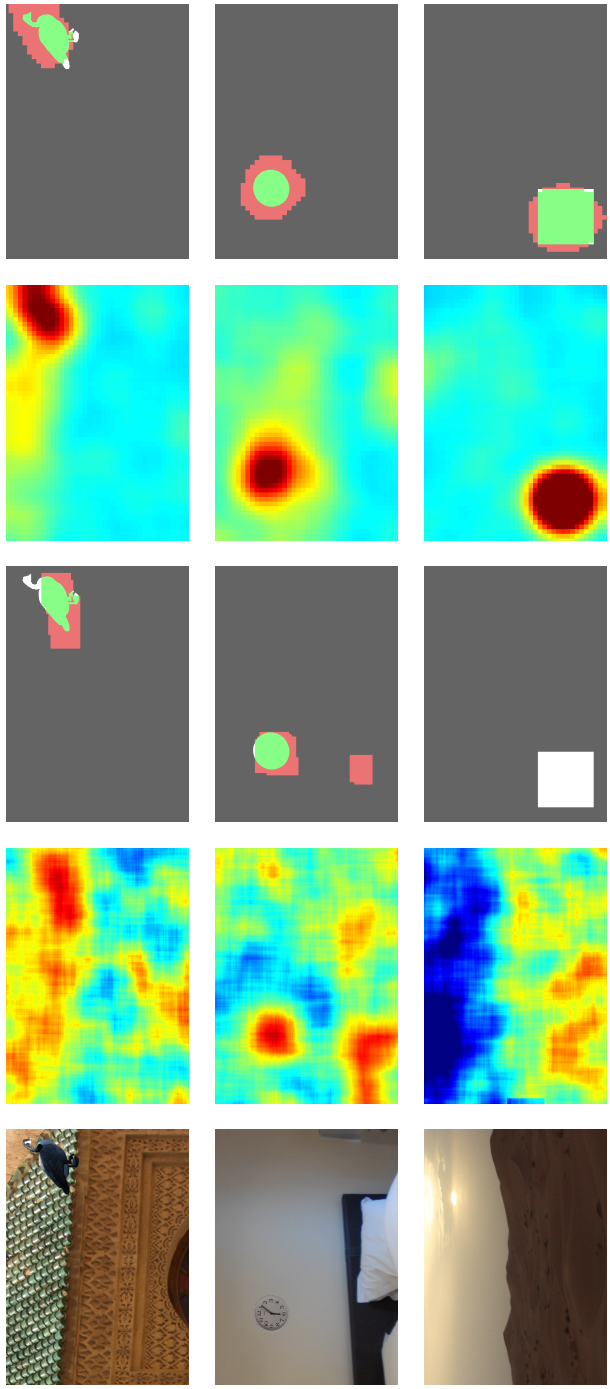


Figure 3.7: Forgery localization results for some selected examples. From left to right: forged image, PRNU correlation index field and color-coded detection mask, proposed aggregation map and color-coded detection mask. Green indicates correct detection, false alarms are in red.

Conclusion

Digital image forensics is a relatively new field of research, with a growing number of contributions from researchers of different background, but also with a large number of issues still open.

In this thesis, we face the problems of image forgery detection and localization through passive techniques. A general solution to these problems does not exist, and a large number of techniques have been proposed in literature following various approaches. As explained in the introduction, the availability of different and complementary tools is essential to obtain satisfactory results in real-world situations, where there is a wide typology of possible manipulations.

Here, we have provided contributions in three different contexts: PRNU-based approaches, pixel-based copy-move detection, and model-based methods. In the first case, we have developed a fast strategy that exploits the image structure, through guided filtering, to improve the localization of small-size forgeries. Future developments will include a segmentation-based analysis. In the second context, we have devised a novel dense-field technique for copy-move forgery detection, based on a fast patch matching tool recently proposed in the literature. We obtained a significant improvement in terms of localization accuracy and speed compared with the state-of-the-art, with good robustness to copy-move rotation and other types of distortion. A future goal is to improve robustness also with respect scale changes and other geometric transformations. In the third scenery, we adopted a model-based method relying on a suitable local image descriptor. Despite its low complexity, this technique provides a very good performance in a wide range of conditions. In future research we will consider a blind scenery, in which the camera is not known.

All techniques have been carefully validated by experimenting on large and meaningful datasets, and comparing performance with the state-of-the-art references. In all cases, besides performance, we paid great attention to computational complexity. Moreover, we take pride in making our research

reproducible, by publishing online source or executable code and reference datasets.

There are several open questions, of a more general nature, for future research. A first one is how to merge the many available tools [30, 42] in order to obtain reliable results for the widest possible range of attacks. Also, the extension of these methods to the case of video poses new problems, not only for the different nature of the data, but also for the largely increased computational load. Another interesting scenario arises when an expert attacker, aware of the specific forensic approach, can take countermeasures. Counter-forensics is the research field that studies forensic techniques to find their weak points [12], of interest both for the attacker and the defender in a foreseeable arms race. All such issues are stimulating intense research, making of image forensics one of the more stimulating and competitive topics in the image processing field.

Appendix A

The First IFS-TC Image Forensics Challenge

In last few years, *Digital image forensics* is gaining a great deal of attention in the scientific community. Therefore in 2013 the IEEE Information Forensics and Security Technical Committee (IFS-TC) launched a detection and localization forensics challenge, the First Image Forensics Challenge [3]. It had different goals:

- to provide the community with an open data set and protocol for evaluation of the latest forensics techniques to identify forgeries in digital images;
- to evaluate the current state-of-the-art techniques with respect to their ability to detect and localize forgeries;
- to set forth a standardization protocol as a common comparison ground truth for new techniques.

The challenge comprised several original images captured from different digital cameras with various scenes either indoor or outdoor. No information was provided on the number and types of cameras. The forged images comprised a set of different manipulation techniques such as copy-move and splicing with different degrees of photorealism.
Moreover the challenge included two sections, the phase 1 for image forgery detection and the phase 2 devoted to image forgery localization. For the phase 1, each participating team had to detect the forged images from a test set of

5713 images. The score was defined as

$$S = \frac{\Pr(\widehat{F}|F) + \Pr(\widehat{P}|P)}{2} \tag{A.1}$$

with $P[F]$ indicating the event "image pristine[fake]" and $\widehat{P}[\widehat{F}]$ the event "decision pristine[fake]", respectively. For the phase 2, the teams were required to provide a mask corresponding to the areas declared forged for each image of a test set of 700 forged images. A score was then computed as the average F-measure over all test images. Note that F-measure of a single image is defined as

$$\text{FM} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FN} + \text{FP}} \tag{A.2}$$

where TP (true positive), FN (false negative), and FP (false positive) count, respectively, the number of detected forged pixels, undetected forged pixels, and wrongly detected genuine pixels. A training set was also available, comprising about 1500 images (450 fake and 1050 pristine). The groups participating in the Challenge had the opportunity to receive a limited feedback by submitting their results once a day. Scores were then computed on a randomized subset of the test set to avoid disclosing valuable information through the system.

he GRIP research group of the University Federico II participated in both phases of the Challenge. In this Appendix we describe the adopted strategy [31, 32]. Given the nature of the dataset, we realized very soon that a fusion of different tools was necessary. Indeed, real-world image forgery detection and localization can be extremely challenging because of the wide availability of powerful photo-editing tools which allow for different types of manipulations, and considering the large variety of operative conditions encountered in practice, including compression, blurring, distortions, etc. No single method can be expected to work satisfactorily in all these cases, and in fact the literature confirms [30, 42] that a suitable fusion of tools can largely improve detection performance over single methods, especially in adverse and unpredictable conditions. Therefore, we developed three complementary tools based, respectively, on machine-learning, block matching and camera sensor noise and carried out eventually a suitable fusion of decisions.

In the next three subsections we describe the three proposed tools. Then we describe the decision fusion strategy and provide some numerical results.

A.1 Tool based on machine-learning

Several feature-based techniques have been proposed in the last decade for splicing detection. Major efforts have been devoted to find good statistical models for natural images in order to select the features that guarantee the highest discriminative power. Interestingly, the method proposed in [94] was inspired by prior work carried out in steganalysis which, despite the obvious differences with respect to the forgery detection field, pursues a very similar goal, that is, detecting seemingly invisible alterations of the natural characteristics of an image. Good results are obtained with features based on some co-occurrence matrices computed on the thresholded prediction-error image, also called *residual image*. In fact, modeling the residuals rather than the pixel values is very sensible in these low-level methods (not based on image semantic), since the image content does not help detecting local alterations and should be suppressed altogether. In the context of forgery detection, in particular, considering that splicing typically introduces sharp edges, it is reasonable to characterize statistically some *edge image*, which can also be the output of a simple high-pass filter, like a derivative of first order. As a further advantage, the residual image has a much narrower dynamic range than the original one, allowing for a compact and robust statistical description by means of co-occurrences.

The processing path outlined above, already proposed in [114], can be therefore summarized in the following steps

1. computation of the high-pass residuals;
2. truncation and quantization;
3. feature extraction based on co-occurrence matrices of selected neighbors;
4. design of a suitable classifier on the training set.

Given its compelling rationale, and the promising results obtained in the literature, we chose to adhere strictly to this path. Even so, a large number of design choices had to be made, beginning from the high-pass filter, to end with the classifier, which impact heavily on the performance and require a lengthy development and testing phase. Fortunately, we could rely on the precious results described in a recent work on steganalysis [44], where a large number of models have been considered, analyzed, and made available online to the re-

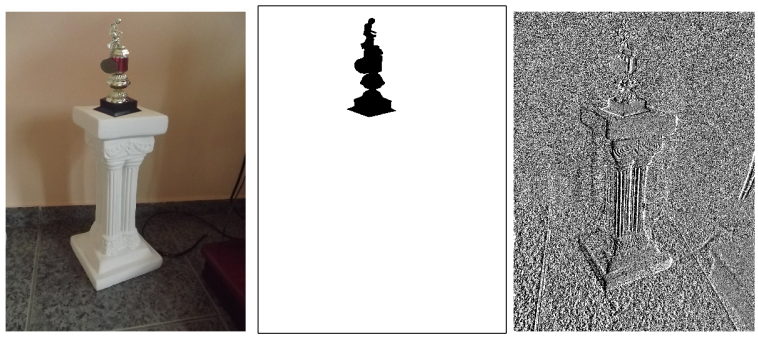


Figure A.1: A training image with its ground truth and an example residual image.

search community [1]. In [44] 39 different high-pass filters are proposed, which work on the grayscale version of the original image obtained by standard conversion. All such filters are quite simple, since their goal is to highlight minor variations w.r.t. typical behaviors. Two examples among the simplest are the first order horizontal *linear* filter

$$r_{i,j} = x_{i,j+1} - x_{i,j}$$

and the first order symmetric *nonlinear* filter

$$r_{i,j} = \min[(x_{i,j+1} - x_{i,j}), (x_{i+1,j} - x_{i,j})]$$

Fig.A.1 shows the effect of applying one of such filters to a training image of the challenge.

Residuals are in general real-valued and, although typically small, are defined on a wide range. To enable their meaningful characterization in terms of co-occurrence they must be quantized and truncated. Following [44] we used

$$\widehat{r}_{ij} = \mathrm{trunc}_T(\mathrm{round}(r_{ij}/q))$$

with q the quantization step and T the truncation value. We used $T = 2$ to limit the matrix size and considered exclusively $q = 1$, both to reduce complexity, and to limit the risk of overfitting to our training set. Each quantized residual can eventually take on 5 values, from -2 to +2. We then computed co-occurrences on four consecutive pixels along the same row or column, obtaining 625 entries, which can be highly reduced thanks to symmetries.

[1] `http://dde.binghamton.edu/download/feature_extractors/`

A.1.1 Detection based on machine-learning

In the classification phase we departed significantly from the reference technique, due to the overfitting problem. In fact, each individual model comprises 169 features for linear filters and 325 for non linear ones, a number large but still manageable with the training set available in the challenge. Merging all models, however, would lead to a much larger number of features, too large to carry out a meaningful training. In [44] this problem was dealt with by means of an *ensemble* classifier, but the training set was about ten times larger.

For the phase 1 of the challenge, we decided therefore to test each model individually, relying heavily on cross validation to gain a reasonable insight into their actual performance. In each experiment, we selected at random 5/6 of the pristine images and 5/6 of the fake ones to train a SVM classifier. The remaining images of each class were then used to test the trained classifier. To reduce randomness, each experiment was repeated 18 times, selecting the training and test set at random, and results were eventually averaged. Fig.A.2(top) shows the results for the 39 models considered, in terms of expected score of the challenge. For several models the predicted score was in the order of 94%, hence very promising. To further improve results, we tried to merge the features of a limited number of models, up to four, not to exceed the number of training images. Results are reported in Tab.A.1 in terms of score obtained before and after merging. The merging did not seem to guarantee any improvement over the best single-model classifier, moreover, the score exhibited a non-monotonic behavior as more models were merged, ringing an alarm bell on stability.

To improve robustness, we considered a different measure of performance. For each SVM classifier, we displaced the separating hyperplane along the orthogonal direction, and built the corresponding ROC. Then we computed, for each model, the Area Under the receiver operating Curve (AUC), because a large AUC implies not only a good performance in the best operating point, but also robustness w.r.t. changing conditions. Fig.A.2(bottom) shows results. Although there is a clear correlation, the top-score models do not coincide with the top-AUC models. We then tried merging the best models selected with this latter criterion, obtaining the results reported in Tab.A.2. This time, performance improved monotonically with merging, providing a gain of about 1% over the best individual model,

Eventually, our SVM classifier used the merging of all the features of models 17, 31, 34 and 36, and was trained over the whole training set.

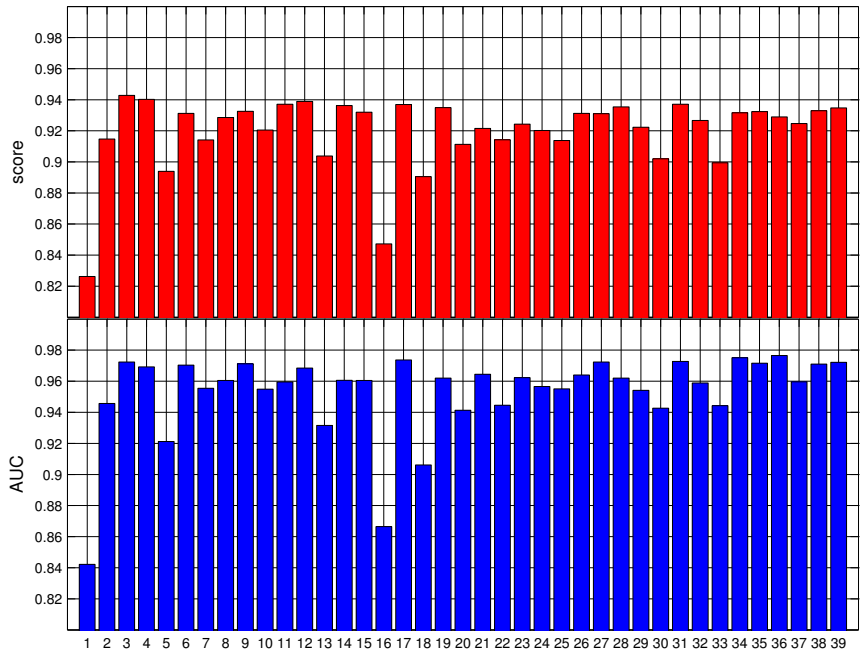


Figure A.2: Scores (top) and AUC (bottom) for all models.

Model	Type	Score	AUC	Score/merg.
3	NL, 1st order	0.9429	0.9724	0.9429
4	NL, 1st order	0.9403	0.9693	0.9154
12	NL, 2nd order	0.9389	0.9685	0.9415
11	NL, 2nd order	0.9371	0.9595	0.9163

Table A.1: Score obtained by the top-score individual models, and by their merging.

A.1.2 Localization based on machine-learning

Given the good performance obtained in the phase 1 of the Challenge we implemented the same procedure in the phase 2 but on a sliding-window basis. Hence, for each image block, the algorithm performed a classification step, collecting not only the sign of the decision, but also its real-valued strength, which measures reliability. Then all strengths were aggregated with their sign to make the final decision.

Model	Type	Score	AUC	Score/merg.
36	linear, 3rd order	0.9289	0.9765	0.9289
34	linear, 1st order	0.9316	0.9751	0.9462
17	NL, 3rd order	0.9369	0.9736	0.9481
31	NL, square 5× 5	0.9371	0.9727	0.9531

Table A.2: Score obtained by the top-AUC individual models, and by their merging.

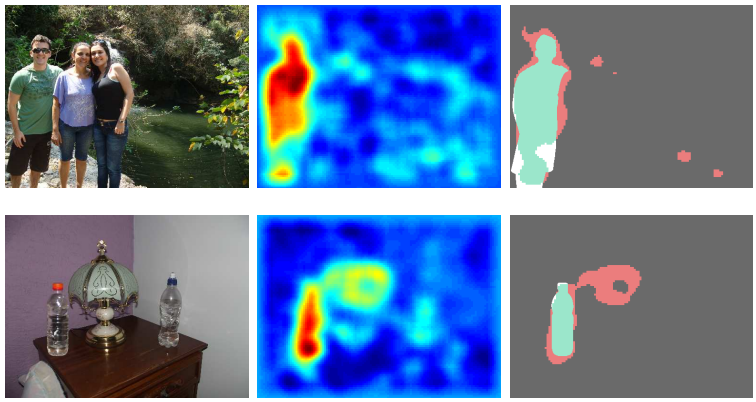


Figure A.3: Two training fake images, their SDH map and the color coded detection mask. Green indicates correct detection, false alarms are in red.

In order to perform classification a preliminary feature extraction process was required, followed by the training of a SVM classifier with linear kernel. Features were computed on 10000 128×128-pixel blocks, 5000 pristine and 5000 fake, extracted by the training images. Note that, in this context, a fake block is not a block drawn entirely from a tampered region, but rather a boundary block, since the most relevant information to discover a forgery is hidden in the transition areas. More precisely, we labeled as fake only the blocks which, according to the ground truth, comprise from 20% to 80% forged pixels. Features were then derived based on the co-occurrence matrices computed on the thresholded prediction-error image using only the best model found in phase 1, a 3rd order linear filter [44].

The image under test was analyzed in sliding-window modality, with partially overlapping 128×128-pixels blocks and a 16-pixel step. For each block

we computed the distance of the corresponding feature vector from the SVM hyperplane, since the larger the distance, the more reliable the result. By aggregating all these values for each pixel we obtained an index, named SDH (Sum of Distances from the Hyperplane), which is roughly related to the probability that the pixel has been tampered. The final binary map was obtained by thresholding this index. Given the peak value of the SDH over the whole image, called PSDH, an empirical analysis on the training set suggested to use a threshold equal to 0.25*PSDH. Fig.A.3 shows some sample results. Note that the PSDH index computed for the this technique represents a measure of detection reliability, useful to guide the final decision fusion. Small values of PSDH can be attributed to random fluctuations, and the corresponding localization map is scarcely reliable.

A.2 Tool based on block-matching

As explained in the Chapter 2, many algorithms for copy-move forgery detection and localization have been proposed in the literature [29]. These techniques aim to discover identical or very similar regions of the image which are likely the effect of some image tampering.

In the Forensics Challenge we used an embryonal version of the *dense* technique described in Chapter 2. The *dense* methods compute a nearest neighbor field (NNF) over all blocks of the image. After this, the areas with homogeneous displacement are selected as candidate forged regions and some candidates are eliminated to reduce false alarms.

We followed a similar line of work, resorting to PatchMatch [9], an iterative algorithm recently proposed for image editing applications. Patchmatch provides a very accurate and regular NNF, but we chose it primarily for its rapid convergence, which makes it about 100 times faster than exact methods, allowing us to process in reasonable time a large database of images. We used 7×7 pixel patches, a size that guarantees a good compromise among accuracy, resolution and speed. All image pixels were preliminarily adjusted to unitary norm, in order to single out copy-moves also in the presence of some intensity adjustments. After computing the NNF, we carried out a filtering on both horizontal and vertical components of the NNF to identify regions with homogeneous displacement. Choosing an appropriate prediction filter, we could also identify regions where displacement vectors slowly increase or decrease linearly, thus identifying also copy-moves with moderate resizing.

All matches obtained in perfectly flat areas, as in presence of saturation,

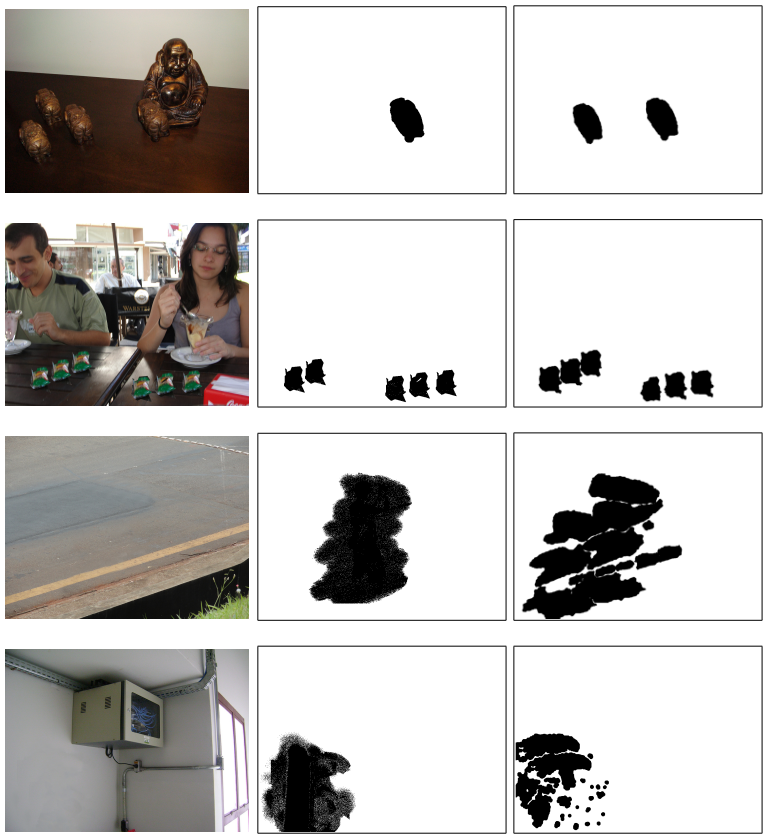


Figure A.4: Four training images with copy-move forgeries, their ground truth, and detection maps output by our method.

were removed to reduce false alarms; likewise, very small regions were also deleted automatically through morphological filtering. For the phase 1, the image was classified as fake if at least one duplicated region was detected. Instead for the phase 2, for each motion vector we compared the image with its shifted version and computed a dense correlation map which, after thresholding and morphological operations, provided the binary map relative to a single copied object. To find also rotated or resized copy-moves, we simply repeated the procedure for a number of rotations and resizing of the image, taking advantage of PatchMatch speed.

Fig.A.4 shows four images with copy-move forgeries, the corresponding ground truth, and the detection map output by our method. Note that the

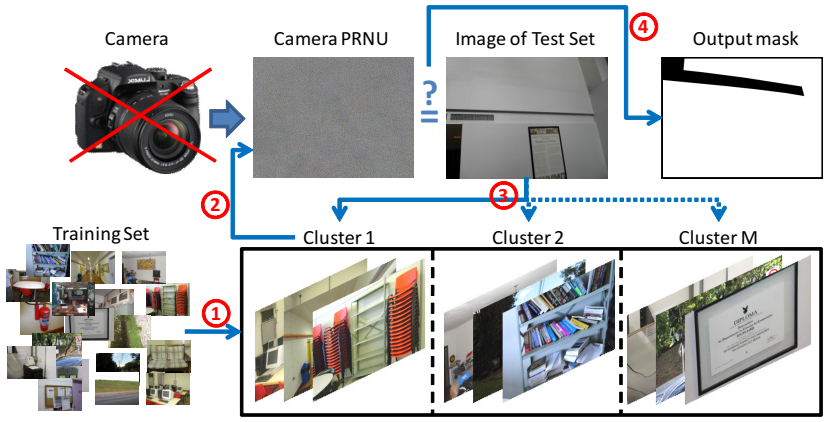


Figure A.5: Steps adopted in PRNU-based tool.

forgery is easily detected, and the map is quite accurate, even when original and copied regions are partially overlapping.

A.3 Tool based on camera sensor noise

The Photo Response Non Uniformity (PRNU) noise represents sort of a camera fingerprint, which is present in all pristine images produced by the camera but absent in tampered areas. By detecting the presence/absence of the camera PRNU in the image under test, one is able to make reliable decisions on the presence of forgeries [77, 27]. A more detailed description of the PRNU-based approach is given in Chapter 1. In general, the camera PRNU pattern is assumed to be already available, but this is only true if we have a collection of images taken by the camera large enough to carry out a reliable estimate. However, this is not always the case, and certainly not the case of the Challenge, since no information was available on the origin of either the training or the test images. Indeed, a large number of images were available, but no information was disclosed about the cameras used to take them. In principle, each of these images could have been taken by a different camera, frustrating any attempt to use a PRNU-based strategy. However, we relied on the reasonable conjecture that the unknown number of cameras M used to build the database was much smaller than N.

Our algorithm comprises therefore the following steps:

1. group the training images in $C + 1$ clusters (one for left-overs);
2. estimate the PRNU for the C valid clusters;
3. associate each test image with one of the clusters;
4. localize forgeries.

For the clustering, we relied on the fact that two images, taken by the same camera, have the same PRNU pattern. At the end of the first step, clusters formed by a sufficient number of images would allow us to estimate the corresponding camera PRNU and perform forgery detection. To carry out the clustering we used the algorithm proposed in [15] which is a simplified version of the well-known pairwise nearest neighbor (PNN) algorithm. In PNN, at the beginning each data vector v_j is the center of a cluster with just one element and weight $w_j = 1$. Then, the two closest centers, say v' and v'' are merged together, provided they are closer than a given threshold, generating by weighted averaging a new center that replaces the existing ones, in formulas

$$\begin{aligned} v_{\text{new}} &= (w'v' + w''v'')/(w' + w'') \\ w_{\text{new}} &= w' + w'' \end{aligned} \tag{A.3}$$

By so doing, the number of centers decreases by one at a time, and the process continues until all centers are farther apart than the threshold, providing the desired clustering. Even fast versions of PNN, however, are computationally demanding, as distances among all couples of data vectors must be computed. The algorithm proposed in [15] introduces some modifications to reduce computation time, like picking at random couples to be compared with the threshold, or looking for all points of a cluster before proceeding with another one.

In our case, the data vectors represent basic estimates of the camera PRNU that are gradually improved through merging. As distance measure we used the Peak to Correlation Energy ratio (PCE) [49], more robust than the correlation index. Since the images used in the challenge might have been cropped at random from larger images produced by the cameras (image dimensions vary from 480×640 to 3240×4320), we had to consider the correlation of an image w.r.t. all shifted versions of another one, and pick the maximum. This was accomplished, as in [50], by first zero-padding images to the same size and then working in the transform domain, obtaining at once the distances corresponding to all circularly shifted versions. We carried out the clustering on the

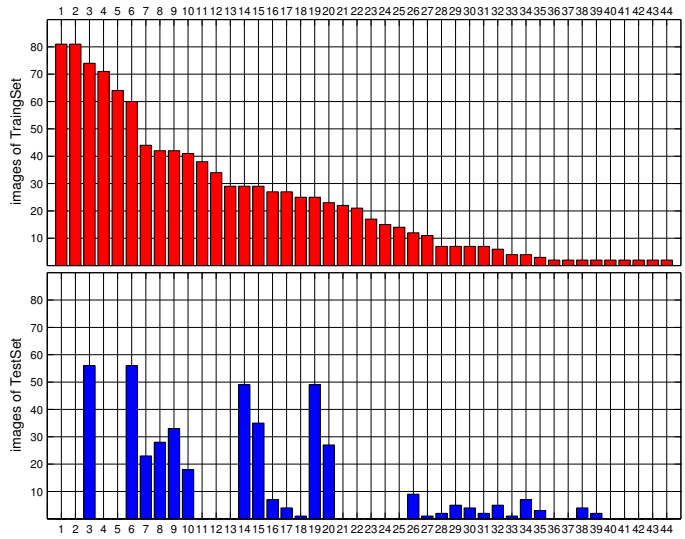


Figure A.6: Number of images belonging to the clustered sets.

training set using a threshold equal to 50, and identified 44 different clusters, for a total of 746 pristine images out of the 1050 available and 315 fakes out of 450 (see Fig.A.6).

Although in the clustering phase the PRNU was estimated, the final estimate for the cluster C was computed according to the maximum likelihood rule [77]. At this point we tried to associate the test images with one of the estimated PRNU's. Setting a threshold equal to 100 on the PCE, we were able to classify 431 of the 700 images available, about 60% of the total, as shown in Fig.A.6.

For all forged images belonging to one of the identified clusters, forgery detection was carried out as proposed in [20] using the normalized correlation index to the 129$\times$ 129 window W_i centered on the target pixel. With respect to the original algorithm, there are two main differences. First, to improve the quality of the noise residuals we resorted to nonlocal denoising, in particular to BM3D [37]. This choice, as shown in [24, 27], improves the separation between image content and PRNU, especially in textured areas. In addition, we used an adaptive decision threshold, which depends on the reliability of the correlation field, measured by the PCE. In fact, due to the lack of prior information on the cameras, the correlation fields are not all equally reliable, depending on the estimation accuracy of the reference PRNU (which depends

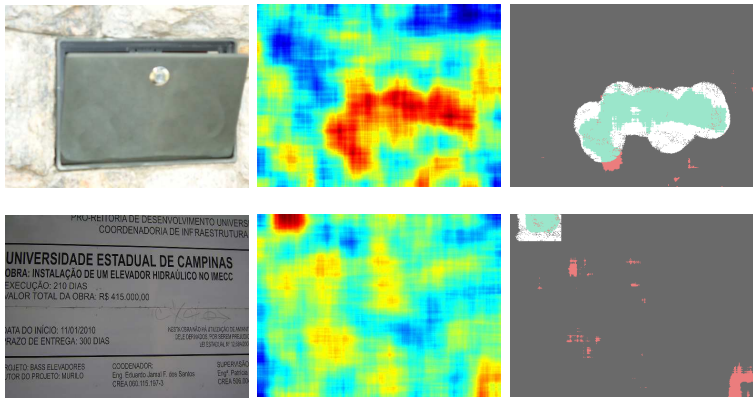


Figure A.7: Two training fake images, correlation maps and color-coded detection masks. Green indicates correct detection, false alarms are in red.

in turn on how many images were available in the cluster) and of the noise residual under test. Therefore, the PCE provides us with a rough measure of reliability, which will be extremely valuable in the decision fusion phase.

It is worth underlining that the correlation might happen to be very low when the image is dark, saturated or strongly textured, increasing the false alarm probability in these areas. In [20] this problem is addresses by means of a predictor which, based on local images features, such as texture, flatness and intensity, computes the expected value of the correlation index under the hypothesis that PRNU is present. In the Challenge, we did not use the predictor, as it proves unreliable when estimated only on a few images. However, we kept enforcing a control on saturated areas, where the PRNU is totally unreliable. In Fig.A.7 we show two images of the training set with the corresponding correlation maps (low values correspond to red in this case) and detection masks.

The algorithm for copy-move forgeries is not able to distinguish the original object from the copy. However, we can use the information coming from the PRNU-based approach (when available) to remove this uncertainty as in the example of Fig.A.8. This technique, however, worked well only when the tested objects were relatively large and the correlation map was sufficiently reliable (PCE$>$150), in all other cases we declared both regions as forged.

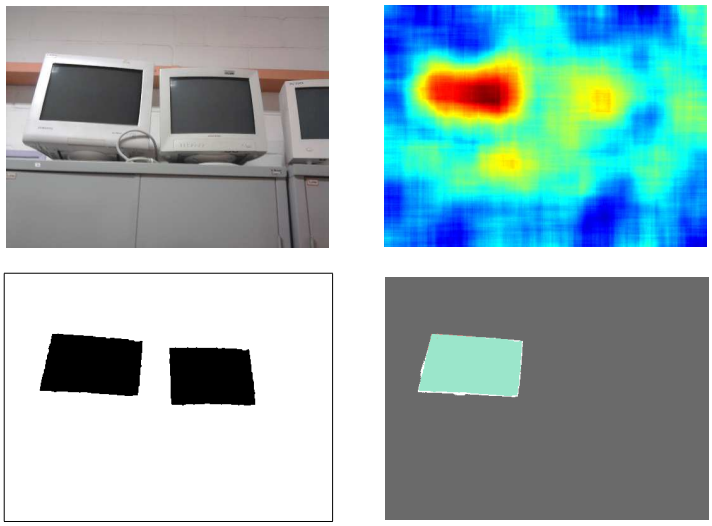


Figure A.8: A training fake image, its correlation map, its PatchMatch-based map, and the final color-coded mask.

A.4 Decision fusion

We implemented three tools based on quite different approaches, machine-learning, block matching, and sensor noise. For the forgery detection, the tool based on machine-learning guaranteed an excellent performance on the training set, with a missing detection rate of 7.10%, and a false alarm rate of 2.29%. Moreover, the "fake" decision of copy-move detector was very reliable, it detected the large majority of the copy-move forgeries in the training set with only 5 false alarms out of 1050 pristine images. Instead, the PRNU-based tool had poor detection performance.

Given these premises, for the phase 1 we discarded the PRNU-based tool and our fusion rule consisted in a simple OR among only two tools: an image was declared fake whenever any of the tools did so, and pristine only if both tools agreed on that.

For the phase 2, our fusion strategy is described by the flow-chart of Fig.A.9. A general guideline was to keep into great account all information about reliability. In particular, since F-measure results computed on the training set made very clear the superior reliability of the copy-move detector, we used only its map when available, and integrated it with the PRNU-based map only when the latter was itself extremely reliable (PCE>1200). Then when

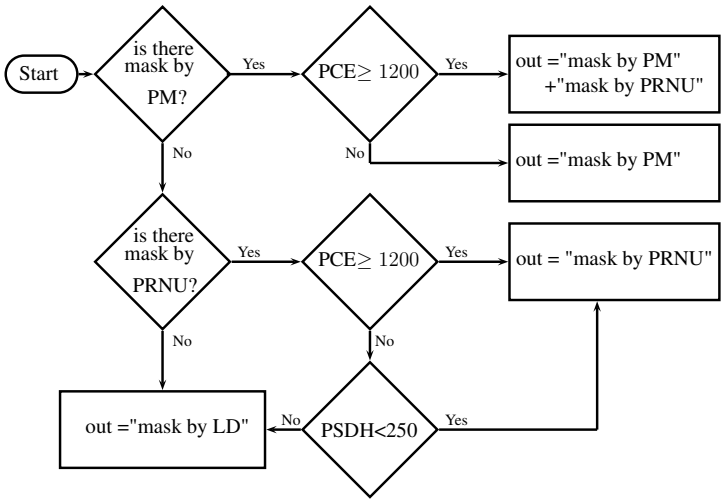


Figure A.9: Flow chart of the combination strategy.

no copy-move was detected, we trusted, in decreasing order, the PRNU-based map and the map based on machine-learning.

#	Leader	Team	Score
1	Luisa Verdoliva	grip	0.9421
2	Guanshuo Xu	havefun	0.9373
3	Xinqi Lin	hyrup	0.9346
4	Licong Chen	Chen	0.9323
5	Khosro Bahrami	Fake Bluster	0.8574
6	Dev Sh	ITD	0.8240

Table A.3: Final ranking (first six teams) for phase 1 of the Challenge.

A.5 Results and Conclusions

For phase 1 of the challenge, our final score, computed on the whole test set, was 0.9421 as opposed to the 0.9738 on the training set. Note that the score obtained by running individually the two approaches is 0.8130 for the copy-move detection and 0.9150 for the method based on machine-learning. Interestingly,

#	Leader	Team	FM
1	Luisa Verdoliva	grip	0.4072
2	Guanshuo Xu	havefun	0.2678
3	Licong Chen	Chen	0.1843
4	Xinqi Lin	hyrup	0.1643

Table A.4: Final ranking for phase 2 of the Challenge.

Method	FM
PRNU	0.1620
LD	0.1115
PM	0.3425
SIFT [6]	0.0528
Zernike [29]	0.1609
JPEG [13]	0.0418
Demosaicking [40]	0.1013

Table A.5: Comparisons of the three approaches described in this work with some state-of-the-art methods.

the scores of the first four groups, shown in Tab.A.3, were very close to one another suggesting that a plateau had been probably reached.

For phase 2, on the training set our strategy provided an average F-measure of 0.4153, and a very similar result was obtained on the test set, 0.4072 (Table A.4). Four sample results on the test set are shown in Fig.A.10.

In order to have a better insight on the experimental results, we also ran individually the three methods and compared them with some approaches appeared in the recent literature [6, 29, 13, 40]. As it is possible to observe from Table A.5 the score obtained by the three tools used in the Challenge are comparatively good. The described strategies allowed us to rank first in both phases of the Challenge.

We feel there are quite a few lessons to learn from this experience. Under a strictly technical point of view, exploring locally the statistical features in the images is arguably the state-of-the-art approach in forgery detection. Nonetheless, the fusion of more tools can further improve performance. In the future research we will focus especially on the fusion of the available information,

both at the pixel and image level, by a more effective strategy [30, 42, 41] than the empirical rules used in this contest. Under a wider point of view, we believe that the Challenge [3] , with its large corpus of images and well-defined performance evaluation protocols, represents an important step for the growth of this field.

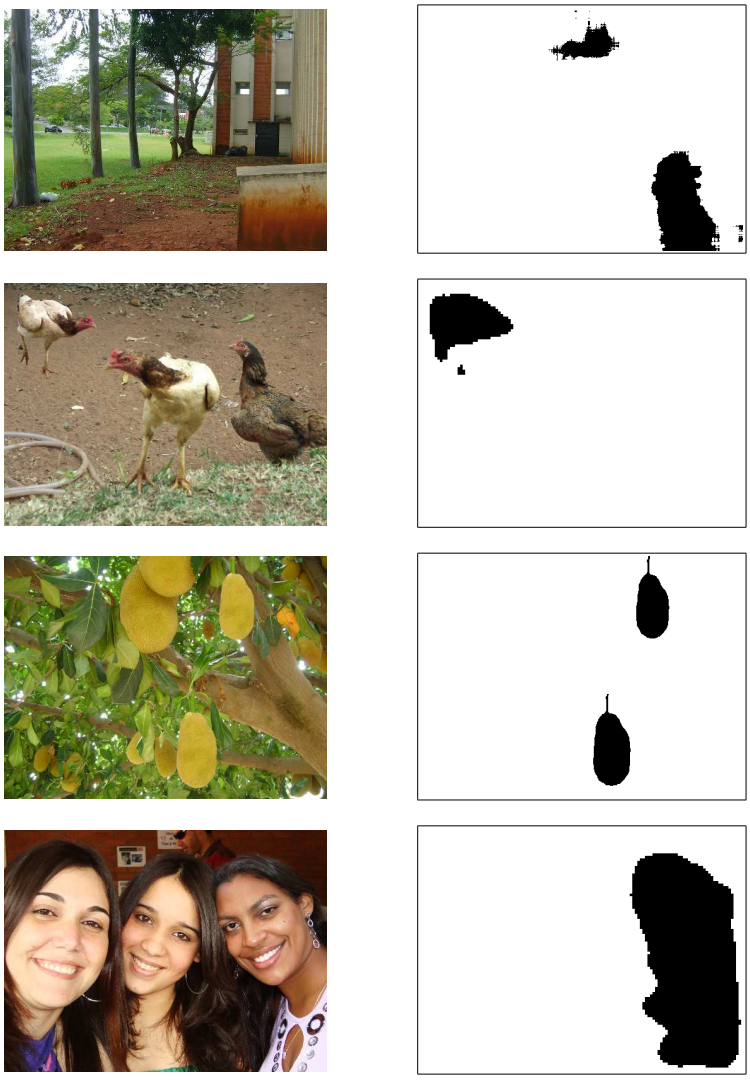


Figure A.10: Four images from the test set and their output masks.

Bibliography

[1] Celebrities before-after Photoshop. [Online]. Available: http://viralscape.com/celebrities-before-after-photoshop/

[2] Photo tampering throughout history. [Online]. Available: http://www.fourandsix.com/photo-tampering-history/

[3] The first Image Forensics Challenge. [Online]. Available: http://ifc.recod.ic.unicamp.br/fc.website/

[4] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[5] O.-M. Al-Qershi and B. E. Khoo, "Passive detection of copy-move forgery in digital images: State-of-the-art," *Forensic Science International*, vol. 231, no. 1-3, pp. 284–295, 2013.

[6] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, L. D. Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with j-linkage," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 659–1669, 2013.

[7] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra, "A sift-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.

[8] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, *Locality-Sensitive Hashing Scheme Based on p-Stable Distributions*. MIT Press, 2006.

[9] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 24:1–24:11, Jul. 2009.

[10] C. Barnes, E. Shechtman, D. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *European Conference on Computer Vision (ECCV)*, vol. 6313. Springer Berlin Heidelberg, 2010, pp. 29–43.

[11] S. Bayram, H. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr. 2009, pp. 1053–1056.

[12] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics*. Springer New York, 2013, pp. 327–366.

[13] T. Bianchi, A. De Rosa, and A. Piva, "Improved dct coefficient analysis for forgery localization in jpeg images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May. 2011, pp. 2444–2447.

[14] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of jpeg artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, Jun. 2012.

[15] G. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 532–534, Mar. 2008.

[16] S. Bravo-Solorio and A. K. Nandi, "Automated detection and localisation of duplicated regions affected by reflection, rotation and scaling in image forensics," *Signal Processing*, vol. 91, no. 8, pp. 1759–1770, 2011.

[17] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind identification of source cell-phone model," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, Sep. 2008.

[18] C. Chan, M. Tahir, J. Kittler, and M. Pietikäinen, "Multiscale Local Phase Quantization for Robust Component-Based Face Recognition Using Kernel Fusion of Multiple Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1164–1177, 2013.

[19] L. Chen, W.-. Lu, J. Ni, W. Sun, and J. Huang, "Region duplication detection based on harris corner points and step sector statistics," *J. Vis. Commun. Image R.*, vol. 24, pp. 244–254, 2013.

[20] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[21] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of jpeg images via periodicity analysis of compression artifacts for tampering detection," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 396–406, Jun. 2011.

[22] Z. Chen, Y. Zhao, and R. Ni, "Forensics of blurred images based on no-reference image quality assessment," in *IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP)*, Jul. 2013, pp. 437–441.

[23] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014, pp. 6231–6235.

[24] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in prnu based forgery detection," in *proc. of the 2Nd ACM Workshop on Multimedia in Forensics, Security and Intelligence*. ACM, 2010, pp. 117–122.

[25] G. Chierchia, S. Parrilli, G. Poggi, L. Verdoliva, and C. Sansone, "Prnu-based detection of small-size image forgeries," in *International Conference on Digital Signal Processing (DSP)*, Jul. 2011, pp. 1–6.

[26] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "Prnu-based forgery detection with regularity constraints and global optimization," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2013, pp. 236–241.

[27] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A bayesian-mrf approach for prnu-based image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 554–567, Apr. 2014.

[28] V. Christlein, C. Riess, and E. Angelopoulou, "On rotation invariance in copy-move forgery detection," in *IEEE International Workshop on Information Forensics and Security*, Dec. 2010.

[29] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[30] D. Cozzolino, F. Gargiulo, C. Sansone, and L. Verdoliva, "Multiple classifier systems for image forgery detection," in *Image Analysis and Processing (ICIAP)*, ser. Lecture Notes in Computer Science, A. Petrosino, Ed. Springer Berlin Heidelberg, 2013, vol. 8157, pp. 259–268.

[31] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5297–5301.

[32] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5302–5306.

[33] D. Cozzolino, S. Parrilli, G. Scarpa, G. Poggi, and L. Verdoliva, "Fast adaptive nonlocal sar despeckling," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 2, pp. 524–528, Feb. 2014.

[34] D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "A Comparative Analysis of Forgery Detection Algorithms," in *International Workshops on Statistical techniques in Pattern Recognition (SPR), LNCS 7626*. Springer Berlin Heidelberg, 2012, pp. 693–700.

[35] D. Cozzolino, G. Poggi, and L. Verdoliva, "Copy-Move forgery detection based on PatchMatch," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5312–5316.

[36] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Transactions on Information Forensics and Security*, submitted.

[37] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[38] C.-A. Deledalle, L. Denis, G. Poggi, F. Tupin, and L. Verdoliva, "Exploiting patch similarity for sar image processing: The nonlocal paradigm," *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 69–78, Jul. 2014.

[39] H. Farid, "Image forgery detection," *IEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[40] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[41] M. Fontani, E. Argones-Rua, C. Troncoso, and M. Barni, "The watchful forensic analyst: Multi-clue information fusion with background knowledge," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov. 2013, pp. 120–125.

[42] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on dempster-shafer theory of evidence," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593–607, Apr. 2013.

[43] J. Fridrich, "Sensor defects in digital image forensic," in *Digital Image Forensics*. Springer New York, 2013, pp. 179–218.

[44] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[45] J. Fridrich, D. Soukal, and J. Lukás, "Detection of copy-move forgery in digital images," in *proc. of Digital Forensic Research Workshop*, 2003.

[46] H. Fu and X. Cao, "Forgery authentication in extreme wide-angle lens using distortion cue and fake saliency map," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1301–1314, Aug. 2012.

[47] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *proc. of the 25th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 518–529.

[48] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in PRNU-based camera identification: a 'Dresden Image Database' case-study," in *proc. of the on Multimedia and security*. ACM, 2012, pp. 109–114.

[49] M. Goljan, "Digital camera identification from images - estimating false acceptance probability," in *Digital Watermarking*. Springer Berlin Heidelberg, 2009, vol. 5450, pp. 454–468.

[50] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *proc. SPIE*, vol. 6819, 2008, pp. 0E–13.

[51] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "An investigation of local descriptors for biometric spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 849–863, Apr. 2015.

[52] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Local contrast phase descriptor for fingerprint liveness detection," *Pattern Recognition*, vol. 48, no. 4, pp. 1050–1058, Apr. 2015.

[53] J.-M. Guo, Y.-. Liu, and Z.-J. Wu, "Duplication forgery detection using improved daisy descriptor," *Expert Systems with Applications*, vol. 40, pp. 707–714, 2013.

[54] Z. Guojuan and L. Dianji, "An overview of digital watermarking in image forensics," in *Fourth International Joint Conference on Computational Sciences and Optimization (CSO)*, Apr. 2011, pp. 332–335.

[55] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[56] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognition*, vol. 45, no. 12, pp. 4292–4299, 2012.

[57] G. Healey and R. Kondepudy, "Radiometric ccd camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, Mar. 1994.

[58] Y.-N. Hsu, H. Arsenault, and G. April, "Rotation-invariant digital pattern recognition using circular harmonic expansion," *Applied Optics*, vol. 21, no. 22, pp. 4012–4015, Nov. 1982.

[59] T. Jing, X. Li, and F. Zhang, "Image tamper detection algorithm based on Radon and Fourier-Mellin transform," in *IEEE International Conference on Information Theory and Information Security (ICITIS)*, Dec. 2010, pp. 212–215.

[60] M. Johnson and H. Farid, "Metric measurements on a plane from a single image," Department of Computer Science, Dartmouth College, Tech. Rep, Tech. Rep. TR2006-579, 2006.

[61] M. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 450–461, Sep. 2007.

[62] P. Kakar and N. Sudha, "Exposing postprocessed copy-paste forgeries through transform-invariant features," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1018–1028, Jun. 2012.

[63] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *proc. of the ACM Workshop on Multimedia and Security*. ACM, 2008, pp. 11–20.

[64] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *SPIE, Electronic Imaging, Media Forensics and Security*, vol. 7541, 2010, pp. 10–12.

[65] S. Korman and S. Avidan, "Coherency sensitive hashing," in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 1607–1614.

[66] M. Kutner, C. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*. McGraw-Hill, 2004.

[67] A. Langille and M. Gong, "An efficient match-based duplication detection algorithm," in *The 3rd Canadian Conference on Computer and Robot Vision*, Jun. 2006, pp. 64–64.

[68] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, Jun. 2010.

[69] C.-T. Li and Y. Li, "Color-decoupled photo response non-uniformity for digital image forensics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 260–271, Feb. 2012.

[70] L. Li, S. Li, H. Zhu, and X. Wub, "Detecting copy-move forgery under affine transforms for image forensics," *Computers & Electrical Engineering*, vol. 40, no. 6, pp. 1951–1962, 2014.

[71] Y. Li, "Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching," *Forensic Science International*, vol. 224, no. 1-3, pp. 59–67, 2013.

[72] S. Liao and M. Pawlak, "On the accuracy of zernike moments for image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1358–1364, Dec. 1998.

[73] C.-Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767–782, May 2001.

[74] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.

[75] B.-B. Liu, Y. Hu, and H.-K. Lee, "Source camera identification from significant noise residual regions," in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2010, pp. 1749–1752.

[76] Q. Liu, X. Cao, C. Deng, and X. Guo, "Identifying image composites through shadow matte consistency," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1111–1122, Sep. 2011.

[77] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *proc. of the SPIE*, vol. 6072, 2006, pp. 720Y–11.

[78] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic Science International*, vol. 171, pp. 180–189, 2007.

[79] F. Marra, F. Roli, D. Cozzolino, C. Sansone, and L. Verdoliva, "Attacking the triangle test in sensor-based camera identification," in *IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5307–5311.

[80] M. Mihçak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6, Mar. 1999, pp. 3253–3256.

[81] A. Mittal, A. Moorthy, and A. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[82] A. Mittal, R. Soundararajan, and A. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[83] G. Muhammad, M. Hussain, and G. Bebis, "Passive copy move image forgery detection using undecimated dyadic wavelet transform," *Digital Investigation*, vol. 9, no. 1, pp. 49–57, 2012.

[84] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Feb. 2009, pp. 331–340.

[85] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[86] I. Olonetsky and S. Avidan, "Treecann - k-d tree coherence approximate nearest neighbor algorithm," in *Europeanl Conference on Computer Visio (ECCV)*. Springer Berlin Heidelberg, 2012, vol. 7575, pp. 602–615.

[87] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 857–867, Dec. 2010.

[88] A. Piva, "An overview on image forensics," *ISNR Signal Processing*, pp. 1–22, Oct. 2012.

[89] G. Poggi, D. Cozzolino, and L. Verdoliva, "Self-organizing maps for the design of multiple description vector quantizers," *Neurocomputing*, vol. 122, pp. 298–309, 2013.

[90] A. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, Oct. 2005.

[91] S.-J. Ryu, M. Kirchner, M.-J. Lee, and H.-K. Leee, "Rotation invariant localization of duplicated image regions based on zernike moments," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1355–1370, Aug. 2013.

[92] S.-J. Ryu, M.-J. Lee, and H.-K. Lee, "Detection of copy-rotate-move forgery using zernike moments," in *Information Hiding*, vol. 6387. Springer Berlin Heidelberg, 2010, pp. 51–65.

[93] Y. Sheng and H. Arsenault, "Experiments on pattern recognition using invariant fourier–mellin descriptors," *Journal of the Optical Society of America*, vol. 3, no. 6, pp. 771–776, Jun. 1986.

[94] Y. Shi, C. Chen, G. Xuan, and W. Su, "Steganalysis versus splicing detection," in *International Workshop on Digital Watermarking*. Springer Berlin Heidelberg, 2008, vol. 5041, pp. 158–172.

[95] B. Shivakumar and S. Baboo, "Detection of region duplication forgery in digital images using surf," *International Journal of Computer Science*, vol. 8, no. 4, pp. 199–205, 2011.

[96] Y. Sutcu, B. Coskun, H. Sencar, and N. Memon, "Tamper detection based on regularity of wavelet transform coefficients," in *IEEE International Conference on Image Processing (ICIP)*, vol. 1, Sep. 2007, pp. 397–400.

[97] M. Teague, "Image analysis via the general theory of moments$*$," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, Aug. 1980.

[98] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *IEEE computer society conference on Computer vision and pattern recognition.*, vol. 2, 2003, pp. II–691.

[99] L. Verdoliva, D. Cozzolino, and G. Poggi, "A feature-based approach for image tampering detection and localization," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 1755–1760.

[100] Q. Wu, S. Wang, and X. Zhang, "Detection of image region-duplication with rotation and scaling tolerance," in *Computational Collective Intelligence. Technologies and Applications LNCS*, vol. 6421, 2010, pp. 100–108.

[101] Q. Wu, S. Wang, and X. Zhang, "Log-polar based scheme for revealing duplicated regions in digital images," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 559–652, 2011.

[102] Y. Xin, M. Pawlak, and S. Liao, "Accurate computation of zernike moments in polar coordinates," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 581–587, Feb. 2007.

[103] L. F. Y. Cao, T. Gao and Q. Yang, "A robust detection algorithm for copy-move forgery in digital images," *Forensic Science International*, vol. 214, pp. 33–43, Jan. 2012.

[104] W. S. Y. Huang, W. Lu and D. Long, "Improved dct-based detection of copy-move forgery in images," *Forensic Science International*, vol. 3, pp. 178–184, 2011.

[105] P.-T. Yap, X. Jiang, and A. Kot, "Two-dimensional polar harmonic transforms for invariant image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1259–1270, Jul. 2010.

[106] I. Yerushalmy and H. Hel-Or, "Digital image forgery detection based on lens and sensor aberration," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 71–91, 2011.

[107] H.-D. Yuan, "Blind forensics of median filtering in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1335–1345, Dec. 2011.

[108] C. Zhang and H. Zhang, "Exposing digital image forgeries by using canonical correlation analysis," in *International Conference on Pattern Recognition (ICPR)*, Aug. 2010, pp. 838–841.

[109] J. Zhao and J. Guo, "Passive forensics for copy-move image forgery using a method based on dct and svd," *Forensic Science International*, vol. 233, no. 1-3, pp. 158–166, 2013.

[110] J. Zhao and W. Zhao, "Passive forensics for region duplication image forgery based on harris feature points and local binary patterns," *Mathematical Problems in Engineering*, vol. 2013, p. 12 pages, 2013.

[111] X. Zhao, S. Wang, S. Li, J. Li, and Q. Yuan, "Image splicing detection based on noncausal markov model," in *IEEE International Conference on Image Processing (ICIP)*, Sep. 2013, pp. 4462–4466.

[112] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using zernike moments and local features," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 55–63, Jan. 2013.

[113] L. Zhu, A. Rao, and A. Zhang, "Theory of keyblock-based image retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 2, pp. 224–257, 2002.

[114] D. Zou, Y. Shi, W. Su, and G. Xuan, "Steganalysis based on markov model of thresholded prediction-error image," in *IEEE International Conference on Multimedia and Expo*, Jul. 2006, pp. 1365–1368.

[115] P. E. Zwicke and I. Kiss, "A new implementation of the mellin transform and its application to radar classification of ships," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 191–199, Mar. 1983.